Web Security

Mục tiêu tấn công đa dạng: Kẻ xấu tấn công web không chỉ để trộm dữ liệu hay tiền bạc, mà còn để tống tiền, làm bàn đạp tấn công khác, khoe khoang, phá hoại, hoặc làm mất uy tín công ty.
Cảnh báo quan trọng: Kiến thức và công cụ bảo mật rất mạnh mẽ. Sử dụng chúng để tấn công hệ thống mà không được phép là vi phạm pháp luật và đạo đức (biến hacker thành cracker), dễ bị phát hiện qua logs và có thể bị truy tố. Sự cho phép (permission) là yếu tố then chốt.
Các dịch vụ dễ bị tấn công: Mail (SMTP), RPC, chia sẻ file Windows (NetBIOS), FTP, SSH, và đặc biệt là Web Server (HTTP/HTTPS) thường là mục tiêu do có nhiều lỗ hổng tiềm ẩn.
Quy trình tấn công Web Server: Thường gồm các bước: Quét tìm cổng mở -> Xác định dịch vụ/phiên bản (banner grabbing) -> Phân tích hệ điều hành (fingerprinting) -> Tìm điểm yếu cụ thể (lỗ hổng, cấu hình sai) -> Khai thác lỗ hổng.
OWASP Top 10: Là danh sách 10 loại lỗ hổng bảo mật web nghiêm trọng và phổ biến nhất cần đặc biệt chú ý, bao gồm:
Không kiểm tra đầu vào (Unvalidated Parameters)
Sai sót kiểm soát truy cập (Broken Access Control)
Quản lý tài khoản/phiên yếu kém (Broken Account/Session Management)
Lỗi XSS (Cross-Site Scripting)
Tràn bộ đệm (Buffer Overflows)
Chèn lệnh (Command Injection - gồm SQL Injection)
Xử lý lỗi kém (Error Handling)
Sử dụng mật mã yếu (Poor Cryptography)
Lỗi giao diện quản trị (Remote Admin Flaws)
Cấu hình server sai (Misconfiguration)
Nguyên tắc bảo mật cơ bản: Tắt dịch vụ thừa, vá lỗi thường xuyên, không tin đầu vào, kiểm tra logic nghiệp vụ, hạn chế lộ thông tin, bảo vệ dữ liệu nhạy cảm (mã hóa, kiểm soát truy cập).
Công cụ thường dùng: Các công cụ như WebGoat (môi trường thực hành), VMWare (máy ảo), nmap (quét mạng), Wireshark (bắt gói tin), Metasploit (khai thác), Brutus (dò pass)... được dùng để kiểm thử và minh họa.
Chốt lại: Bảo mật web là cuộc chiến liên tục, cần hiểu rõ mục tiêu và phương pháp của kẻ tấn công, nhận biết các lỗ hổng phổ biến (OWASP Top 10), và áp dụng các nguyên tắc bảo mật cơ bản trong suốt quá trình phát triển và vận hành. Luôn nhớ hành động có trách nhiệm và đúng pháp luật.

The goal of an attack

- Steal data
- Blackmail
- Beachhead for other attacks
- Bragging rights
- Vandalism
- Demonstrate vulnerability/satisfy curiosity
- Damage company reputation

Steal data (Đánh cắp dữ liệu):
Đây là mục tiêu rất phổ biến. Kẻ tấn công muốn lấy các thông tin nhạy cảm như: thông tin thẻ tín dụng, thông tin cá nhân người dùng (tên, địa chỉ, số điện thoại, email), mật khẩu, bí mật kinh doanh, mã nguồn, dữ liệu khách hàng...
Dữ liệu này có thể được bán trên chợ đen, sử dụng để lừa đảo, hoặc tống tiền.
Blackmail (Tống tiền):
Kẻ tấn công có thể đánh cắp dữ liệu nhạy cảm và đe dọa công khai nó trừ khi nạn nhân trả tiền chuộc.
Hoặc họ có thể chiếm quyền kiểm soát hệ thống (ví dụ: mã hóa dữ liệu - ransomware) và đòi tiền chuộc để khôi phục lại.
Beachhead for other attacks (Bàn đạp/Đầu cầu cho các cuộc tấn công khác):
"Beachhead" là một thuật ngữ quân sự, nghĩa là chiếm một vị trí nhỏ ban đầu để làm bàn đạp mở rộng tấn công sâu hơn vào lãnh thổ đối phương.
Trong bảo mật web, điều này có nghĩa là kẻ tấn công chiếm quyền kiểm soát một máy chủ web (có thể không quá quan trọng) để làm điểm xuất phát, từ đó tấn công vào các hệ thống quan trọng hơn bên trong mạng nội bộ (như cơ sở dữ liệu, máy chủ ứng dụng khác...). Hoặc dùng máy chủ đã chiếm được để gửi thư rác, phishing, hoặc làm một phần của mạng botnet.
Bragging rights (Để khoe khoang/Lấy tiếng):
Một số kẻ tấn công (đặc biệt là những người trẻ tuổi hoặc muốn thể hiện) tấn công các hệ thống nổi tiếng chỉ để khoe khoang kỹ năng của mình trong cộng đồng hacker. Việc hack thành công một trang web lớn mang lại "danh tiếng".
Vandalism (Phá hoại):
Tấn công chỉ đơn giản để phá hoại, làm thay đổi giao diện trang web (deface), xóa dữ liệu, hoặc làm gián đoạn dịch vụ mà không có mục đích tài chính rõ ràng. Đôi khi vì mục đích chính trị (hacktivism) hoặc chỉ đơn giản là muốn phá.
Demonstrate vulnerability/satisfy curiosity (Chứng minh lỗ hổng/ Thỏa mãn sự tò mò):
Kẻ tấn công tìm thấy một lỗ hổng bảo mật và tấn công chỉ để chứng minh rằng lỗ hổng đó tồn tại, hoặc đơn giản là vì tò mò muốn xem họ có thể xâm nhập vào hệ thống hay không. Đôi khi họ không có ý định gây hại nghiêm trọng sau khi đã vào được.
Damage company reputation (Phá hủy danh tiếng công ty):
Tấn công nhằm mục đích làm tổn hại đến hình ảnh và uy tín của một công ty. Việc website bị hack, dữ liệu khách hàng bị lộ, hoặc dịch vụ bị gián đoạn có thể gây mất lòng tin nghiêm trọng từ phía khách hàng và đối tác. Đôi khi đây là mục tiêu của đối thủ cạnh tranh.

A word of warning

"These tools and techniques can be dangerous" (Các công cụ và kỹ thuật này có thể nguy hiểm):
Các công cụ dùng để kiểm tra bảo mật (quét cổng, dò lỗ hổng, bẻ khóa mật khẩu...) cũng chính là những công cụ mà kẻ xấu (cracker) sử dụng để tấn công.
Sử dụng chúng không đúng cách, dù là vô tình, cũng có thể gây hại cho hệ thống (làm sập dịch vụ, mất dữ liệu).

These tools and techniques can be dangerous

The difference between a hacker and a cracker is…permission

Admins will see strange activity in logs, and come looking for you

Authorities are prosecuting even the “good guys” for using these tools

"The difference between a hacker and a cracker is… permission" (Sự khác biệt giữa hacker và cracker là... sự cho phép):
Đây là một điểm rất quan trọng.
Hacker: Thường được hiểu (theo nghĩa gốc hoặc "mũ trắng") là người có kỹ năng máy tính cao, tò mò, thích khám phá hệ thống, và thường hành động có đạo đức, có sự cho phép (ví dụ: kiểm thử xâm nhập cho công ty mình hoặc khách hàng).
Cracker: Là người sử dụng kỹ năng máy tính để xâm nhập hệ thống một cách trái phép, với ý đồ xấu (trộm cắp, phá hoại ...).
Ranh giới mỏng manh chính là "permission" (sự cho phép).
Làm điều tương tự nhưng không được phép là phạm pháp.

"Admins will see strange activity in logs, and come looking for you" (Admin sẽ thấy hoạt động lạ trong logs, và sẽ tìm đến bạn):
Mọi hành động trên hệ thống mạng (quét cổng , cố gắng đăng nhập, truy cập tệp...) đều thường được ghi lại trong các tệp nhật ký (logs).
Quản trị viên hệ thống (Admins) theo dõi các logs này. Nếu bạn thực hiện các hoạt động trái phép, họ sẽ phát hiện ra và có thể truy tìm nguồn gốc.

"Authorities are prosecuting even the “good guys” for using these tools" (Chính quyền đang truy tố cả những "người tốt" vì sử dụng các công cụ này):
Đây là lời cảnh báo về pháp lý. Việc truy cập trái phép vào hệ thống máy tính là vi phạm pháp luật ở hầu hết các quốc gia.
Ngay cả khi bạn không có ý đồ xấu, chỉ tò mò hoặc "thử nghiệm" mà không được phép, bạn vẫn có thể bị truy cứu trách nhiệm hình sự.
Pháp luật không phân biệt bạn là "người tốt" hay "người xấu" khi bạn hành động trái phép.

Commonly attacked services

SMTP servers (port 25)

- sendmail: “The address parser performs insufficient bounds checking in certain conditions due to a char to int conversion, making it possible for an attacker to take control of the application”

RPC servers (port 111 & others)

NetBIOS shares (ports 135, 139, 445)

- Blaster worm
- Sasser worm

FTP servers (ports 20, 21)

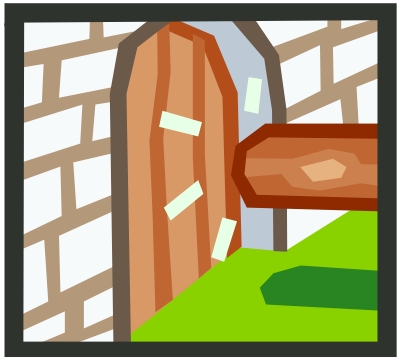
- wuftpd vulnerabilities

SSH servers (port 22)

- OpenSSH, PAM vulnerabilities

Web servers (ports 80, 443)

- Apache chunked encoding vulnerability



SMTP servers (port 25):
Dịch vụ gửi mail (Simple Mail Transfer Protocol).
Ví dụ lỗ hổng: Lỗi tràn bộ đệm (insufficient bounds checking) trong sendmail (một phần mềm máy chủ mail phổ biến) cho phép kẻ tấn công chiếm quyền kiểm soát.
RPC servers (port 111 & others):
Dịch vụ gọi thủ tục từ xa (Remote Procedure Call), cho phép chương trình yêu cầu dịch vụ từ chương trình khác trên máy tính khác. Thường có nhiều lỗ hổng bị khai thác.
NetBIOS shares (ports 135, 139, 445):
Dịch vụ chia sẻ tệp và máy in trong mạng Windows.
Ví dụ mã độc: Các loại worm nổi tiếng như Blaster và Sasser đã khai thác lỗ hổng trong các dịch vụ này để lây lan nhanh chóng.
FTP servers (ports 20, 21):
Dịch vụ truyền tệp (File Transfer Protocol).
Ví dụ lỗ hổng: Các lỗ hổng trong wuftpd (một phần mềm máy chủ FTP).
SSH servers (port 22):
Dịch vụ đăng nhập và thực thi lệnh từ xa an toàn (Secure Shell).
Ví dụ lỗ hổng: Các lỗ hổng trong OpenSSH (phần mềm SSH phổ biến) hoặc PAM (hệ thống quản lý xác thực).
Web servers (ports 80, 443):
Dịch vụ web (HTTP port 80, HTTPS port 443). Đây là mục tiêu cực kỳ phổ biến.
Ví dụ lỗ hổng: Lỗ hổng trong cách xử lý "chunked encoding" của máy chủ web Apache.

Web server attack

Scan to find open ports

Mục đích: Xác định xem máy chủ mục tiêu đang chạy những dịch vụ mạng nào. Mỗi dịch vụ mạng thường "lắng nghe" trên một hoặc nhiều cổng (port) cụ thể (ví dụ: web server thường ở cổng 80/443, FTP ở cổng 21...).

Find out what's running on open ports (banner grabbing)

Mục đích: Sau khi biết cổng nào đang mở, cần xác định chính xác phần mềm (tên, phiên bản) đang chạy dịch vụ trên cổng đó.

Profile the server

- Windows (look for Kerberos, NetBIOS, AD)
- Unix
- Use TCP fingerprinting

Mục đích: Thu thập thêm thông tin chi tiết về hệ điều hành và cấu hình của máy chủ.

Probe for weaknesses on interesting ports

- Default configuration files and settings (e.g. popular IIS ones)
- Buffer overflows
- Insecure applications

Mục đích: Khi đã biết dịch vụ nào đang chạy và hệ điều hành là gì, kẻ tấn công bắt đầu tìm kiếm các lỗ hổng cụ thể.

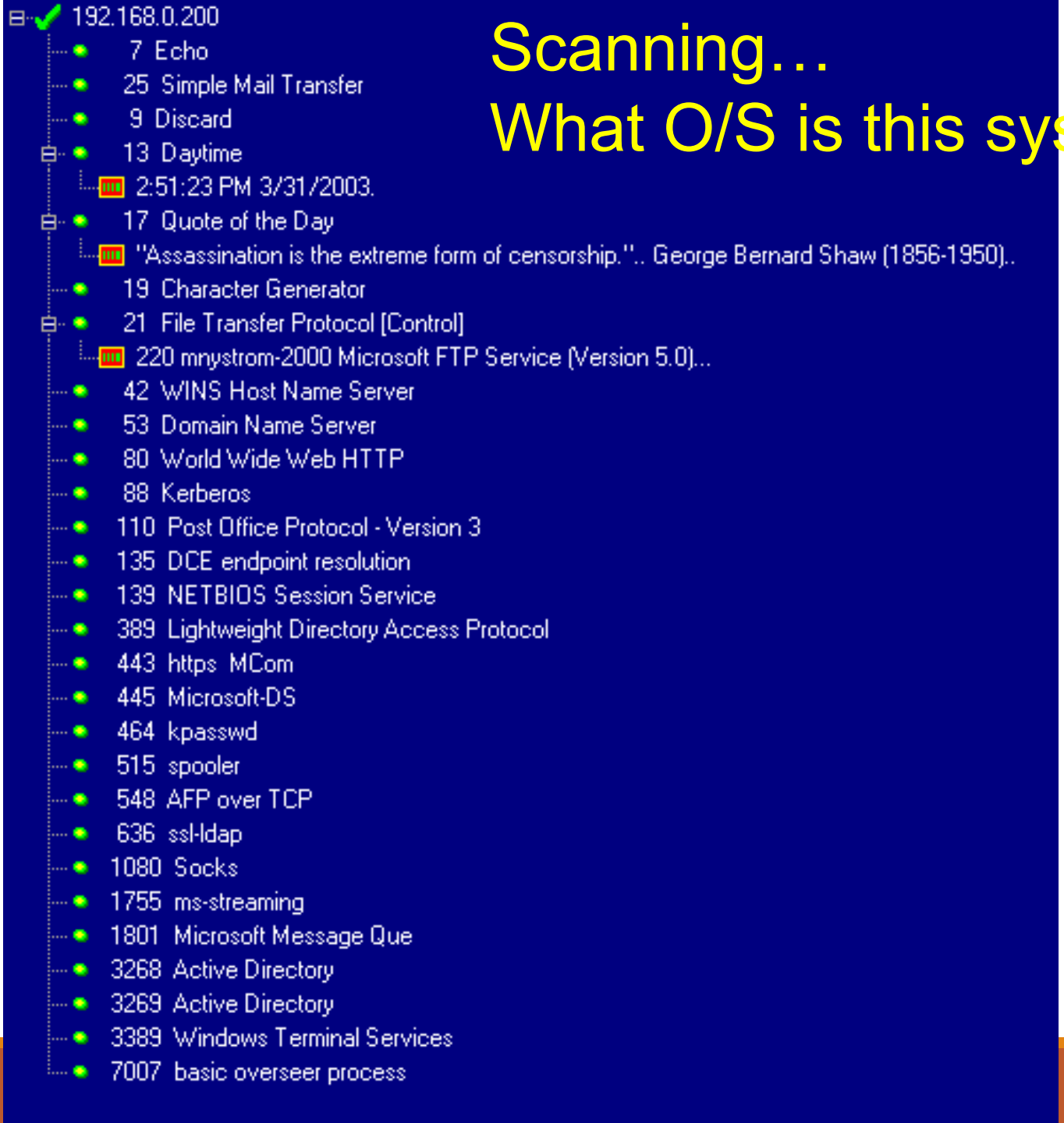
Launch attack

- Use exploit code from Internet...
- ...or build your own

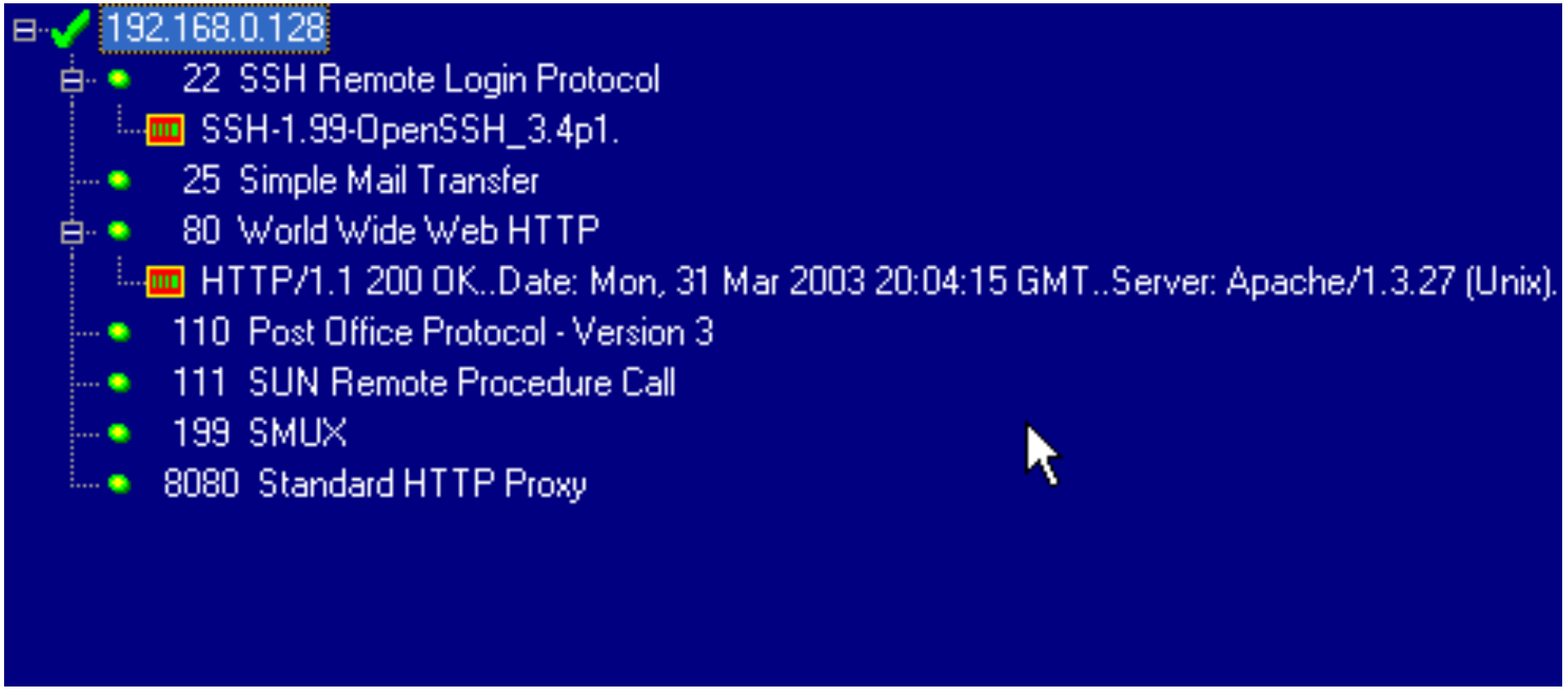
Mục đích: Khai thác lỗ hổng đã tìm thấy để đạt được mục tiêu (chiếm quyền kiểm soát, đánh cắp dữ liệu...).
Cách làm:
Use exploit code from Internet: Tìm kiếm và sử dụng các đoạn mã khai thác (exploit code) có sẵn trên mạng cho lỗ hổng cụ thể đã xác định được (ví dụ: từ Metasploit Framework, Exploit Database).
...or build your own: Nếu là lỗ hổng mới (zero-day) hoặc không có exploit công khai, kẻ tấn công có kỹ năng cao có thể tự viết mã khai thác riêng.

Default configuration files and settings: Kiểm tra xem máy chủ có đang sử dụng cấu hình mặc định hay không (thường kém an toàn), tìm các tệp cấu hình hoặc trang quản trị mặc định (ví dụ: các trang cấu hình mặc định của IIS - web server của Microsoft).
Buffer overflows: Thử gửi dữ liệu quá lớn vào các dịch vụ xem có gây ra lỗi tràn bộ đệm hay không (một loại lỗ hổng phổ biến).
Insecure applications: Kiểm tra các ứng dụng web chạy trên máy chủ (nếu có) xem có các lỗ hổng phổ biến như SQL injection, XSS, Unvalidated Parameters... hay không (đây là trọng tâm của OWASP Top 10).

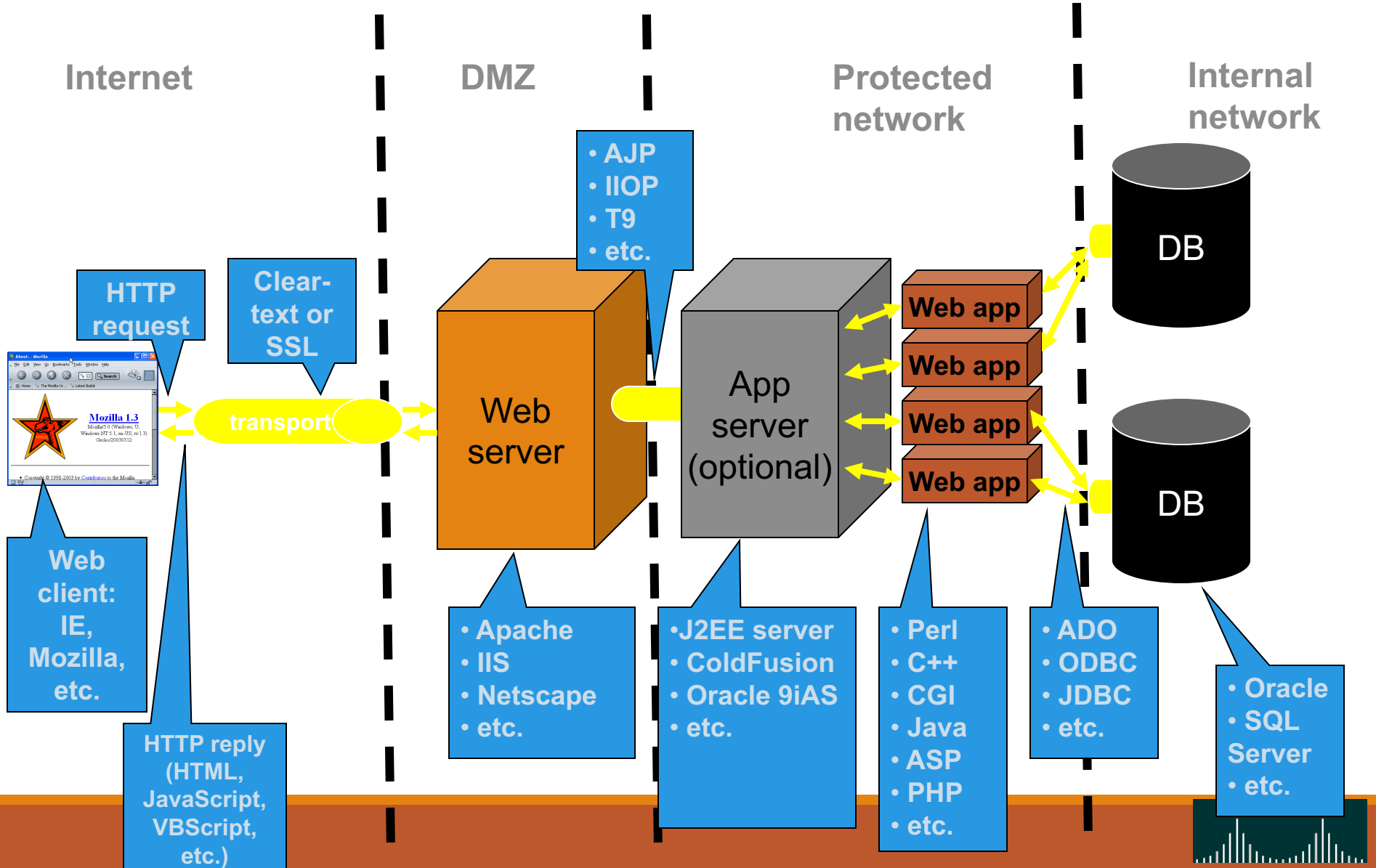
Scanning…
What O/S is this system?



Scanning…
What O/S is this system?



Example Web Application



OWASP Top 10 Web Application Security Vulnerabilities

http://www.owasp.org

1. Unvalidated parameters
2. Broken access control
3. Broken account/session management
4. Cross-site scripting flaws
5. Buffer overflows
6. Command injection flaws
7. Error handling problems
8. Insecure use of cryptography
9. Remote administration flaws
10. Web and app server mis-configuration

OWASP Top 10 là gì?
Đây là một tài liệu nâng cao nhận thức (awareness document) rất nổi tiếng và có ảnh hưởng trong ngành bảo mật.
Nó liệt kê 10 loại rủi ro bảo mật được coi là nghiêm trọng nhất và phổ biến nhất đối với các ứng dụng web, dựa trên sự đồng thuận của các chuyên gia bảo mật trên toàn thế giới.
Danh sách này không phải là tất cả các lỗ hổng có thể có, nhưng nó tập trung vào những cái quan trọng nhất mà các nhà phát triển, quản trị viên và chuyên gia bảo mật cần đặc biệt chú ý.
OWASP Top 10 được cập nhật định kỳ (vài năm một lần) để phản ánh những thay đổi trong bối cảnh mối đe dọa. (Lưu ý: Danh sách trên slide có thể là một phiên bản cũ, nhưng các khái niệm cốt lõi vẫn rất quan trọng).

Unvalidated parameters (Tham số không được xác thực): Ứng dụng không kiểm tra kỹ dữ liệu đầu vào từ người dùng (trong URL, form, cookie...), cho phép kẻ tấn công gửi dữ liệu độc hại.
Broken access control (Kiểm soát truy cập bị lỗi): Người dùng có thể truy cập vào các chức năng hoặc dữ liệu mà họ không được phép (ví dụ: xem thông tin người khác, dùng chức năng admin).
Broken account/session management (Quản lý tài khoản/phiên làm việc bị lỗi): Các vấn đề liên quan đến cách xử lý đăng nhập, mật khẩu, ID phiên, cookie... dẫn đến việc tài khoản hoặc phiên làm việc có thể bị chiếm đoạt.
Cross-site scripting flaws (Lỗi kịch bản chéo trang - XSS): Kẻ tấn công chèn mã độc (thường là JavaScript) vào trang web, mã này sau đó sẽ thực thi trên trình duyệt của người dùng khác, có thể dùng để ăn cắp thông tin hoặc thực hiện hành động trái phép dưới danh nghĩa nạn nhân.
Buffer overflows (Tràn bộ đệm): Gửi quá nhiều dữ liệu vào một vùng nhớ đệm của ứng dụng, làm ghi đè lên vùng nhớ khác, có thể dẫn đến treo ứng dụng hoặc thực thi mã độc. (Phổ biến hơn ở các ứng dụng viết bằng C/C++).
Command injection flaws (Lỗi chèn lệnh): Kẻ tấn công có thể chèn và thực thi các lệnh hệ thống (OS commands) hoặc lệnh cơ sở dữ liệu (SQL injection - một dạng rất phổ biến) thông qua các trường nhập liệu của ứng dụng.
Error handling problems (Vấn đề xử lý lỗi): Ứng dụng hiển thị quá nhiều thông tin chi tiết kỹ thuật trong thông báo lỗi, giúp kẻ tấn công thu thập thông tin về hệ thống.
Insecure use of cryptography (Sử dụng mật mã không an toàn): Lưu trữ dữ liệu nhạy cảm (mật khẩu, thẻ tín dụng) mà không mã hóa đúng cách, sử dụng thuật toán yếu, hoặc quản lý khóa mã hóa kém.
Remote administration flaws (Lỗi quản trị từ xa): Giao diện quản trị ứng dụng/máy chủ không an toàn (mật khẩu yếu, không mã hóa, bị lộ ra ngoài internet).
Web and app server mis-configuration (Cấu hình sai máy chủ web/ứng dụng): Cài đặt không an toàn trên máy chủ web, máy chủ ứng dụng, hoặc cơ sở dữ liệu (dùng tài khoản mặc định, bật dịch vụ không cần thiết, thiếu bản vá...).

#1: Unvalidated Parameters

Vấn đề cốt lõi: Ứng dụng web tin tưởng mù quáng vào dữ liệu do người dùng gửi lên (qua URL, form, cookie, trường ẩn, headers...) mà không kiểm tra kỹ lưỡng trên máy chủ.

Attacker can easily change any part of the HTTP request before submitting

- URL
- Cookies
- Form fields
- Hidden fields
- Headers

Rủi ro: Kẻ tấn công có thể dễ dàng thay đổi bất kỳ phần nào của dữ liệu này trước khi nó được gửi đi (dùng các công cụ đơn giản), nhằm mục đích lừa ứng dụng thực hiện hành động trái phép (ví dụ: xem dữ liệu người khác, thay đổi giá tiền, thực thi mã độc...).
Lầm tưởng: Việc mã hóa (encoding) như Base64 không phải là mã hóa bảo mật (encryption), nó dễ dàng bị giải mã ngược.
Nguyên tắc khắc phục: Phải xác thực (validate) MỌI dữ liệu đầu vào trên MÁY CHỦ (server-side), không được chỉ dựa vào kiểm tra phía trình duyệt (client-side) vì nó có thể bị vô hiệu hóa hoặc bỏ qua.

Encoding is *not* encrypting

- Toasted Spam: http://www.toastedspam.com/decode64

Input must be validated on the server (not just the client).

- CoolCarts: http://www.extremelasers.com

Biện pháp:
Kiểm tra kỹ code (code review), chỉ chấp nhận dữ liệu tốt đã biết (allow list).
Sử dụng tường lửa ứng dụng web (WAF).
Dùng các công cụ quét mã và giám sát.

Countermeasures

- Tainting (Perl)
- Code reviews (check variable against list of allowed values, not vice-versa)
- Application firewalls
  - CodeSeeker: http://www.owasp.org/codeseeker/
  - Real-time auditing: http://www.covelight.com

#2: Broken Access Control

Lỗi: Người dùng truy cập được vào chỗ không được phép (trang admin, dữ liệu người khác) do kiểm tra quyền hạn không nhất quán hoặc sai sót .
Vd: Ép truy cập URL bị cấm, dùng ../ để xem file hệ thống.
Sửa: Kiểm tra quyền tập trung, dùng cơ chế có sẵn, xem lại code.

Usually inconsistently defined/applied

Examples

- Forced browsing past access control checks
- Path traversal
- File permissions – may allow access to config/password files
- Client-side caching

Countermeasures

- Use non-programmatic controls
- Verify access control via central container
- Code reviews

#3: Broken Account and Session Management

Lỗi: Mật khẩu yếu, tên đăng nhập dễ đoán, ID phiên đoán được/bị lộ, cho phép kẻ tấn công chiếm tài khoản/phiên làm việc.
Vd: Đoán/reset pass, ăn cắp cookie session.
Sửa: Mật khẩu mạnh, bỏ tên mặc định, bảo vệ session ID, mã hóa lưu trữ pass.

Weak authentication

- Password-only
- Easily guessable usernames (admin, etc.)
- Unencrypted secrets are sniffable

How to break in

- Guess/reset password
- Have app email you new password
- Sniff or crack password

Backend authentication

- How are database passwords stored?
- Trust relationships between hosts (IP address can be spoofed, etc.)

Countermeasures

- Strong passwords
- Remove default user names
- Protect sensitive files

#4: Cross-Site Scripting (XSS)

Attacker uses trusted application/company to reflect malicious code to end-user

Attacker can “hide” the malicious code

- Unicode encoding

2 types of attacks

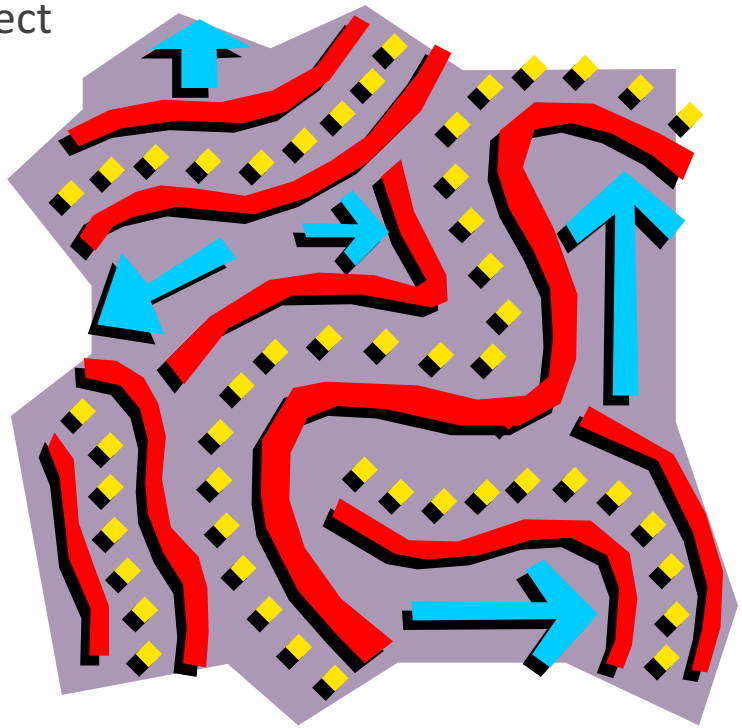
- Stored
- Reflected

Wide-spread problem!

Countermeasures

- input validation
  - Positive
  - Negative: “< > ( ) # &”
  - Don’t forget these: “&lt &gt &#40 &#41 &#35 &#38”
- User/customer education

Lỗi: Kẻ tấn công chèn mã độc (thường là Javascript) vào web, mã này chạy trên trình duyệt người dùng khác khi họ xem trang, dùng để ăn cắp thông tin (cookie...).
Vd: Nhập <script>alert('XSS')</script> vào ô comment.
Sửa: Kiểm tra/lọc/mã hóa (encode) dữ liệu đầu vào/hiển thị (vd: đổi < thành <).



#5: Buffer Overflows

Lỗi: Gửi quá nhiều dữ liệu vào bộ nhớ đệm, làm nó tràn ra ghi đè lên vùng nhớ khác, có thể khiến app sập hoặc bị chiếm quyền. (Hay gặp ở C/C++).
Vd: Gửi 1000 chữ 'a' vào chỗ chỉ chứa được 100.
Sửa: Vá lỗi, xem code, dùng ngôn ngữ an toàn hơn (Java...)

Mostly affects web/app servers

Can affect apps/libraries too

Goal: crash the target app and get a shell

Buffer overflow example

- `echo "vrfy `perl -e 'print "a" x 1000'`" |nc www.targetsystem.com 25`
- Replace all those "a"s with something like this...
- `char shellcode[] = "\xeb\x1f\x5e\x89\x76\x08…"`

Countermeasures

- Keep up with bug reports/patches
- Code reviews
- Run with limited privileges
- Use "safer" languages like Java

#6: Command Injection

Allows attacker to relay malicious code in form variables or URL

- System commands
- SQL
- Interpreted code (Perl, Python, etc.)

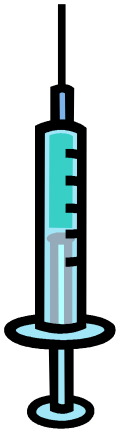
Many apps use calls to external programs

- sendmail

Examples

- Path traversal: “../”
- Add more commands: “; rm –r *”
- SQL injection: “’ OR 1=1”

Lỗi: Kẻ tấn công chèn và thực thi lệnh hệ điều hành hoặc CSDL (SQL Injection) thông qua dữ liệu đầu vào.
Vd: Nhập '; rm -rf /' vào ô tìm kiếm, nhập ' OR 1=1 -- vào ô login.
Sửa: Kiểm tra/lọc đầu vào, dùng tham số hóa truy vấn (parameterized query), không gọi lệnh hệ thống trực tiếp.



Countermeasures

- Taint all input
- Avoid system calls (use libraries instead)
- Run with limited privileges

#7: Error Handling

Lỗi: Thông báo lỗi tiết lộ quá nhiều thông tin kỹ thuật (tên file, cấu trúc DB, stack trace), giúp kẻ tấn công hiểu hệ thống và tìm điểm yếu.
Vd: Hiển thị lỗi SQL chi tiết cho người dùng cuối.
Sửa: Hiển thị thông báo lỗi chung chung cho người dùng, ghi log chi tiết cho admin, tùy chỉnh trang lỗi (404, 500).

Examples: stack traces, DB dumps

Helps attacker know how to target the app

Inconsistencies can be revealing too

- “File not found” vs. “Access denied”

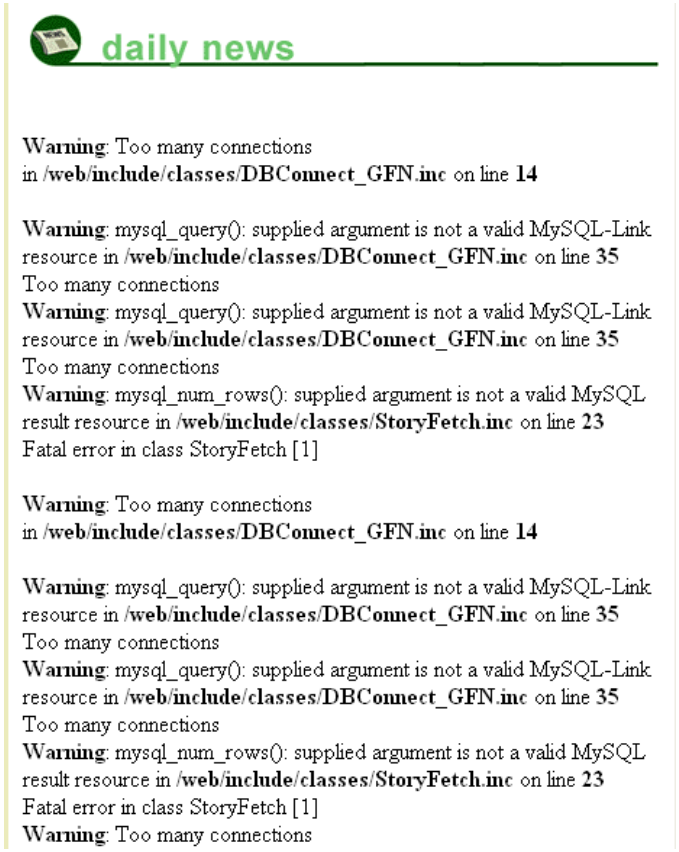
Fail-open errors

Need to give enough info to user w/o giving too much info to attacker

Countermeasures

- Code review
- Modify default error pages (404, 401, etc.)

Error messages example



daily news

Warning: Too many connections
in **/web/include/classes/DBConnect_GFN.inc** on line **14**

Warning: mysql_query(): supplied argument is not a valid MySQL-Link resource in **/web/include/classes/DBConnect_GFN.inc** on line **35**
Too many connections
Warning: mysql_query(): supplied argument is not a valid MySQL-Link resource in **/web/include/classes/DBConnect_GFN.inc** on line **35**
Too many connections
Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in **/web/include/classes/StoryFetch.inc** on line **23**
Fatal error in class StoryFetch [1]

Warning: Too many connections
in **/web/include/classes/DBConnect_GFN.inc** on line **14**

Warning: mysql_query(): supplied argument is not a valid MySQL-Link resource in **/web/include/classes/DBConnect_GFN.inc** on line **35**
Too many connections
Warning: mysql_query(): supplied argument is not a valid MySQL-Link resource in **/web/include/classes/DBConnect_GFN.inc** on line **35**
Too many connections
Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in **/web/include/classes/StoryFetch.inc** on line **23**
Fatal error in class StoryFetch [1]
Warning: Too many connections

Lỗi: Lưu trữ/truyền dữ liệu nhạy cảm không an toàn (pass, thẻ tín dụng), dùng thuật toán yếu, sinh số ngẫu nhiên kém (dễ đoán session ID).
Vd: Lưu pass dạng text, dùng MD5 để hash pass, session ID tăng tuần tự.
Sửa: Chỉ lưu cái cần thiết, hash pass bằng thuật toán mạnh (bcrypt, Argon2), dùng mã hóa chuẩn (HTTPS), không tự chế thuật toán.

#8: Poor Cryptography

Insecure storage of credit cards, passwords, etc.

Poor choice of algorithm (or invent your own)

Poor randomness

- Session IDs
- Tokens
- Cookies

Improper storage in memory

Countermeasures

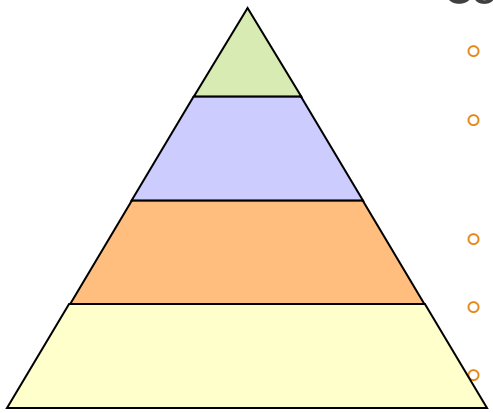
- Store only what you must
- Store a hash instead of the full value (SHA-1)
- Use only vetted, public cryptography

#9: Remote Administration Flaws

Lỗi: Giao diện quản trị web dễ bị tấn công do mật khẩu yếu, không mã hóa, lộ ra ngoài internet.
Vd: Trang login admin dùng pass admin/admin, không có HTTPS.
Sửa: Đặt giao diện admin ở server riêng/cổng khác, xác thực mạnh (2FA, cert), mã hóa (VPN/SSL), giới hạn IP truy cập.

Problems

- Weak authentication (username="admin")
- Weak encryption



Countermeasures

- Don't place admin interface on same server
- Use strong authentication: certificates, tokens, strong passwords, etc.
- Encrypt entire session (VPN or SSL)
- Control who has accounts
- IP restrictions

#10: Web/App Server Misconfiguration

Tension between “work out of the box” and “use only what you need”

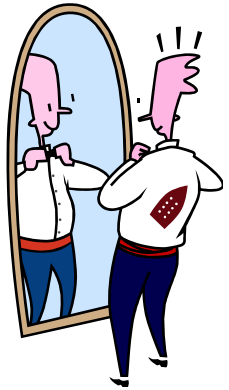
Developers ≠ web masters

Examples

- Unpatched security flaws (BID example)
- Misconfigurations that allow directory traversal
- Administrative services accessible
- Default accounts/passwords

Countermeasures

- Create and use hardening guides
- Turn off all unused services
- Set up and audit roles, permissions, and accounts
- Set up logging and alerts



Lỗi: Cài đặt không an toàn trên server (web, app, DB) do không vá lỗi, bật dịch vụ thừa, dùng tài khoản/mật khẩu mặc định, quyền hạn sai.
Vd: Để lộ trang quản trị server, dùng pass mặc định của DB, cho phép xem nội dung thư mục.
Sửa: Dùng hướng dẫn hardening, tắt dịch vụ thừa, vá lỗi, đổi pass mặc định, kiểm tra quyền hạn, bật logging.

Principles

Turn off un-needed services

Keep systems patched

Don’t trust input

Watch for logic holes

Only provide the necessary information

Hide sensitive information

- Encryption
- Access controls

Turn off un-needed services (Tắt các dịch vụ không cần thiết): Mỗi dịch vụ chạy là một bề mặt tấn công tiềm năng. Nếu không dùng đến, hãy tắt nó đi để giảm thiểu rủi ro.
Keep systems patched (Giữ hệ thống được vá lỗi): Thường xuyên cập nhật hệ điều hành, máy chủ web, cơ sở dữ liệu, và các thư viện/ứng dụng lên phiên bản mới nhất để vá các lỗ hổng bảo mật đã biết.
Don’t trust input (Đừng tin tưởng dữ liệu đầu vào): Nguyên tắc vàng! Luôn kiểm tra, xác thực, và làm sạch (sanitize/escape) mọi dữ liệu đến từ người dùng hoặc các hệ thống bên ngoài trước khi xử lý hoặc lưu trữ. (Liên quan đến #1, #4, #6...).
Watch for logic holes (Để ý các lỗ hổng logic): Ngoài các lỗi kỹ thuật cụ thể, cần xem xét cả quy trình hoạt động của ứng dụng xem có bị lạm dụng về mặt logic hay không (ví dụ: quy trình đặt hàng, chuyển tiền, cấp quyền...).
Only provide the necessary information (Chỉ cung cấp thông tin cần thiết): Tránh hiển thị quá nhiều thông tin không cần thiết cho người dùng, đặc biệt là thông tin chi tiết kỹ thuật trong thông báo lỗi. (Liên quan đến #7).
Hide sensitive information (Che giấu thông tin nhạy cảm): Bảo vệ dữ liệu quan trọng như mật khẩu, thông tin cá nhân, thẻ tín dụng...
Encryption (Mã hóa): Mã hóa dữ liệu khi lưu trữ và truyền đi. (Liên quan đến #8).
Access controls (Kiểm soát truy cập): Đảm bảo chỉ những người được phép mới có thể truy cập thông tin nhạy cảm. (Liên quan đến #2).

Tools used in this preso

WebGoat –vulnerable web applications for demonstration

VMWare – runs Linux & Windows 2000 virtual machines on demo laptop.

nmap –host/port scanning to find vulnerable hosts

Ethereal – network traffic sniffing

Metasploit Framework – exploit tool

Brutus – password cracking

Sleuth – HTTP mangling against web sites