**Tìm hiểu A5 và A6**

1. **Security misconfiguration**

Sai sót cấu hình an ninh: Một cơ chế an ninh tốt cần phải định nghĩa những hiệu chỉnh về an ninh và triển khai nó cho các ứng dụng, khuôn mẫu, máy chủ ứng dụng, máy chủ web, máy chủ dữ liệu và các ứng dụng nền tảng. Tất cả những thiệt lập nên được định nghĩa, thực hiện và bảo trì bởi vì rất nhiều thứ không được triển khai với thiết lập an toàn mặc định. Các hiệu chỉnh cũng bao gồm cập nhật phần mềm và những thư viện được sử dụng bởi ứng dụng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tác nhân đe dọa | Hướng tấn công | Điểm yếu bảo mật | | Tác động kỹ thuật | Ảnh hưởng kinh doanh |
| Đặc trưng ứng dụng | Khả năng khai thác: dễ | Mức độ phổ biến: thông thường | Khả năng phát hiện: dễ | Ảnh hưởng: vừa phải | Đặc trưng ứng dụng/ kinh doanh |
| Có thể là những kẻ tấn công nặc danh từ phía ngoài, hoặc người người dùng sở hữu tài khoản muốn kiểm soát hệ thống. Cũng nên cân nhắc những đối tượng nội bộ muốn che dấu hành động. | Kẻ tấn công sử dụng tài khoản tài khoản mặc định, trang web không được sử dụng, lỗi bảo mật chưa được vá, tập tin và thư mục không được bảo vệ để thu thâp thông tin hoặc truy cập hệ thống một cách bất hợp pháp | Sai sót trong cấu hình an ninh có thể xảy ra ở bất kì cấp độ ứng dụng nào, bao gồm: nền tảng, máy chủ web, máy chủ ứng dụng, các khuôn mẫu hay các đoạn mã tùy chỉnh. Người phát triển và nhà quản trị mạng cần phải làm việc cùng nhau để đảm bảo rằng từng lớp được chỉnh một cách đúng đắn. Những công cụ quét tự động cũng có thể hữu ích trong việc phát hiện những bản vá lỗi bị thiếu, sai sót trong cấu hình hoặc sử dụng những tài khoản mặc định, những dịch vụ không cần thiết. | | Lỗi như vậy thường cho kẻ tấn công quyền truy cập bất hợp pháp đến những chức năng hay dữ liệu hệ thống. Thỉnh thoảng lỗi này cũng có thể giúp kẻ tấn công chiếm toàn bộ hệ thống. | Toàn bộ hệ thống có thể bị chiếm mà bạn không thể biết. Tất cả dữ liệu có thể bị đánh cắp hoặc thay đổi từ từ qua thời gian. Việc sửa chữa sai sót có thể rất tốn kém. |

**Tôi có mắc phải lỗi?**

Bạn đã thực hiện việc bảo vệ an ninh cho từng lớp ứng dụng chưa ?

1. Bạn đã có một quy trình để giữ cho các phần mềm được cập nhật? Bao gồm hệ điều hành, máy chủ web, DBMS (hệ quản lý dữ liệu), ứng dụng, và tất cả các thư viện mã

2. Liệu những thứ không cần thiết có được tắt hoặc xóa đi không ? (cổng mạng, dịch vụ, trang web, tài khoản)

3. Các tài khoản mặc định đã được vô hiệu hay được thay đổi chưa ?

4. Những phần xử lý lỗi có được thiết lập để tránh rò rỉ những dấu vết stack hay những thông tin lỗi quá cụ thể?

5. Bạn có hiểu rõ và cấu hình một cách thích hợp những thiết lập an ninh trong các khuôn mẫu (Ví dụ: Struts, Spring, ASP.Net) và thư viện dùng để phát triển? Nên có một quy trình phối hợp và lặp đi lặp lại để phát triển và duy trì một cấu hình an ninh ứng dụng phù hợp.

**Làm cách nào để chống ?**

Chúng tôi khuyến khích thiết lập tất cả những thứ sau:

1. Một tiến trình bảo mật có thể dễ dàng lặp lại giúp cho việc triển khai trên môi trường khác nhanh chóng và dễ dàng, nên được thiết lập giống nhau, nên được tự động  
để giảm thiểu công sức thiết lập một môi trường mới an toàn

2. Một tiến trình cho việc cập nhật và triển khai tất cả những bản nâng cấp và bản vá của phần mềm một cách định kỳ đối với mỗi môi trường được triển khai. Nó cũng  
bao gồm luôn những thư viện chương trình thường bị bỏ qua.

3. Một kiến trúc ứng dụng vững chắc có thể phân tách và bảo vệ các thành phần riêng biệt

4. Hãy xem xét việc chạy chương trình quét và kiểm tra định kì để phát hiện những cấu hình sai hoặc những bản vá thiếu..

**Ví dụ tình huống tấn công**

Kịch bản 1: Ứng dụng của bạn phụ thuộc vào 1 nền tảng phát triển mạnh như Struct or Spring. Lỗi XSS được phát hiện trong những phần mà bạn phụ thuộc. Một bản vá được phát hành nhưng bạn đã bỏ qua. Kẻ tấn công có thể lợi dụng lỗi đó bất cứ lúc nào nếu bạn còn chưa cập nhật.

Kịch bản 2: Giao diện điều khiển máy chủ của quản trị viên được tự động cài đặt và không được gỡ bỏ. Tài khoản mặc định chưa được thay đổi. Kẻ tấn công phát hiện đăng nhập với tài khoản mặc định và chiếm quyền điều khiển hệ thống.

Kịch bản 3: Chức năng liệt kê thư mục chưa được vô hiệu hóa Kẻ tấn công phát hiện và hắn có thể liệt kê các tập tin trong thư mục và tìm tập tin hắn muốn. Ví dụ những tập tin Java class của bạn, hắn dịch ngược về mã nguồn và tìm ra những lỗ hổng nghiêm trọng trong đó.

Kịch bản 4: Cấu hình máy chủ cho phép người dùng xem các dấu vết stack, điều này có thể để lộ những lỗ hổng bảo mật. Kẻ tấn công rất thích những thông tin được thêm vào trong những thông điệp báo lỗi như vậy

1. **Sensetive Data Explosure**

Lộ dữ liệu nhạy cảm: Nhiều ứng dụng web không bảo vệ dữ liệu nhạy cảm như thẻ tín dụng, SSN, và những mã xác thực thông tin bằng các phương thức mã hóa hay băm (hashing). Kẻ tấn công có thể ăn cắp hay thay đổi những dữ liệu nhạy cảm này và tiến hành hành vi trộm cắp, gian lận thẻ tín dụng, v.v...

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tác nhân đe dọa | Hướng tấn công | Điểm yếu bảo mật | | Tác động kỹ thuật | Ảnh hưởng kinh doanh |
| Đặc trưng ứng dụng | Khả năng khai thác: dễ | Mức độ phổ biến: thông thường | Khả năng phát hiện: dễ | Ảnh hưởng: vừa phải | Đặc trưng ứng dụng/ kinh doanh |
| Hãy xem xét đến những người sử dụng hệ thống của bạn. Liệu họ có muốn truy cập vào dữ liệu mật mà họ không được uỷ quyền không? Quản trị viên nội bộ cũng không ngoại lệ. | Thông  thường, những kẻ tấn công không bẻ mật mã mà phá hoại những thứ khác: chẳng hạn tìm các khóa, lấy các bản sao dữ liệu không được mã hóa, hoặc truy cập dữ liệu qua các kênh có thể tự động giải mã. | Sai lầm phổ biến là không mã hóa các dữ liệu cần được mã hóa. Khi mã hóa được sử dụng, vấn đề lưu trữ và sinh khóa không an toàn, khóa không được thường xuyên thay đổi hay cách sử dụng thuật toán yếu thường xảy ra. Sử dụng mật khẩu dễ đoán hoặc thuật toán băm mã mà không thêm ký tự ngẫu nhiên cũng là một lỗi phổ biến. Kẻ tấn công từ bên ngoài thường gặp khó khăn trong việc phát hiện lỗi do quyền truy cập hạn chế. Chúng thường phải khai thác các khía cạnh khác trước khi đạt được quyền truy cập cần thiết. | | Nhóm lỗ hổng này thường gây ảnh hưởng lên tất cả các dữ liệu cần được mã hóa. Thông thường thông tin này bao gồm các dữ liệu nhạy cảm như hồ sơ y tế, thông tin, dữ liệu cá nhân, thẻ tín dụng, v.v… | Hãy xem xét giá trị kinh doanh của các dữ liệu bị mất và mức độ ành hưởng của những dữ liệu này tới danh tiếng của bạn. Bạn sẽ phải chịu trách nhiệm pháp l{ gì nếu những dữ liệu này bị lộ ra? Đồng thời xem xét các thiệt hại tới danh tiếng của bạn |

**Tôi có mắc phải lỗi?**

Điều đầu tiên bạn phải xác định là thông tin nào nhạy cảm đến mức cần phải yêu cầu bảo mật. Ví dụ: mật khẩu, thẻ tín dụng, hồ sơ y tế, và các thông tin cá nhân nên được mã hóa. Đối với tất cả các dữ liệukiểu như vậy, bảo đảm rằng:

1. Những thông tin được lưu trữ lâu dài phải được mã hóa, đặc biệt trong bản sao lưu các dữ liệu này.

2. Chỉ những người được ủy quyền có thể truy cập các bản sao được giải mã của dữ liệu (Xem A4 và A8).

3. Sử dụng một thuật toán mã hóa mạnh và được xây dựng theo tiêu chuẩn.

4. Một khóa mạnh được tạo và được bảo vệ để tránh những truy cập trái phép. Một kế hoạch thay đổi khóa định kỳ phải được dự trù.

Để có một cái nhìn hoàn chỉnh hơn về các vấn đề cần tránh, xem ASVS requirements on Cryptography (V7)

**Làm cách nào để chống?**

Danh sách đầy đủ các nguy cơ bởi các mật mã không an toàn vượt ra ngoài phạm vi khả năng của văn bản Top 10. Vì vậy để đạt được mục đích an toàn cho tất cả các dữ liệu quan trọng cần được mã hóa, bạn phải ít nhất thực hiện tất cả những điều sau đây:

1. Xem xét các nguy cơ đe dọa tới sự bảo mật của các dữ liệu (ví dụ: tấn công từ bên trong, người sử dụng bên ngoài), chắc chắn bạn mã hóa tất cả dữ liệu còn lạở trạng thái lưu trữ i theo cách thức để bảo vệ chống lại các mối đe dọa này.

2. Các bản sao lưu trữ offsite cũng được mã hóa, chìa khóa của các bản sao lưu này phải được quản lý và sao lưu riêng biệt.

3. Đảm bảo các thuật toán mạnh theo tiêu chuẩn và các khóa mã mạnh được sử dụng, việc quản lý các khóa mã này cũng cần được lưu tâm.

4. Đảm bảo mật khẩu được băm theo tiêu chuẩn của một thuật toán mạnh và có độ phức tạp nhất định.

5. Đảm bảo tất cả các khóa và mật khẩu được bảo vệ khỏi nhũng truy cập trái phép.

**Ví dụ tình huống tấn công**

Kịch bản 1: Mã hóa thẻ tín dụng trong cơ sở dữ liệu được sử dụng để ngăn chặn việc thông tin bị lộ ra với người sử dụng. Tuy nhiên, cơ sở dữ liệu lại được thiết lập để tự động giải mã các truy vấn đối với cột lưu thẻ tín dụng,điều này cho phép kiểu tấn công SQL Injection thu thập được thông tin thẻ tín dụng ở dạng không mã hóa. Thực chất hệ thống này cần phải được thiết lập sao cho những thông tin chỉ được giải mã ở những ứng dụng phía trong, không phải từ những ứng dụng web bên ngoài

Kịch bản 2: Một hồ sơ sức khỏe được mã hóa và được sao lưu lại trên một cuốn băng, nhưng mã khóa của hồ sơ cũng được lưu trên bản sao đó. Cuộn băng đã không bao giờ đến được trung tâm lưu trữ.

Kịch bản 3: một cơ sở dữ liệu về mật khẩu sử dụng unsalted hashes để lưu trữ mật khẩu của mọi người. Một lỗi tải file đã được tận dụng bởi tin tặc để lấy các tập tin mật khẩu. Tất cả unsalted hashes bị giải mã chỉ trong vòng 4 tuần, trong khi đối với những hashes nếu được tạo ra đúng cách (with salt) việc giải mã sẽ phải mất hơn 3000 năm.