ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

Вид нефункционального тестирования, направленный на оценку уязвимости программного обеспечения к различным атакам.

УЯЗВИМОСТЬ

Это недостаток или слабость в проектировании, реализации, эксплуатации или управлении системы, которые могут быть использованы для компрометации целей безопасности системы.

УГРОЗА

Это что угодно (вредоносный внешний злоумышленник, внутренний пользователь, нестабильность системы и т. д.), что может нанести ущерб частям приложения (ценным ресурсам, таким как данные в базе данных или в файлах системы), используя уязвимость.

ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

Это ряд активностей, нацеленных на демонстрацию соответствия приложения требованиям к безопасности, предъявляемым владельцами бизнеса.

OWASP

Open Web Application Security Project

www.owasp.org

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

Должен включать следующие компоненты:

- ✓ Люди - убедиться, что люди обучены и проинструктированы;
- ✓ Процесс – убедиться, что есть адекватные стандарты и стратегии обеспечения качества;
- ✓ Технология – убедиться, что процесс показал себя, как эффективный в процессе его внедрения и выполнения.

ТЕХНИКИ ТЕСТИРОВАНИЯ БЕЗОПАСНОСТИ

- ✓ Ручное инспектирование и ревью;
- ✓ Моделирование угроз;
- ✓ Ревью кода;
- ✓ Тестирование на проникновения.

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

- ✓ Перед началом разработки;
- ✓ Во время создания требований/дизайна;
- ✓ Во время разработки;
- ✓ Во время деплоймента;
- ✓ Поддержка приложения.

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

- ✓ Перед началом разработки;
  - Определение жизненного цикла разработки ПО;
  - Ревью стратегий и стандартов безопасности;
  - Определение параметров и метрик в обеспечении безопасности;
- ✓ Во время создания требований/дизайна;
- ✓ Во время разработки;
- ✓ Во время деплоймента;
- ✓ Поддержка приложения.

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

- ✓ Перед началом разработки;
- ✓ Во время создания требований/дизайна:
  - Ревью требований к безопасности;
  - Ревью архитектуры и дизайна;
  - Создание и ревью UML моделей;
  - Создание и ревью моделей угроз;
- ✓ Во время разработки;
- ✓ Во время деплоймента;
- ✓ Поддержка приложения.

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

- ✓ Перед началом разработки;
- ✓ Во время создания требований/дизайна;
- ✓ Во время разработки:
  - Поверхностный просмотр кода;
  - Код ревью;
- ✓ Во время деплоймента;
- ✓ Поддержка приложения.

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

- ✓ Перед началом разработки;
- ✓ Во время создания требований/дизайна;
- ✓ Во время разработки;
- ✓ Во время деплоймента:
  - Тестирование на проникновения;
  - Тестирование конфигурации приложения;
- ✓ Поддержка приложения.

ЭФФЕКТИВНЫЙ ПРОЦЕСС ТЕСТИРОВАНИЯ

- ✓ Перед началом разработки;
- ✓ Во время создания требований/дизайна;
- ✓ Во время разработки;
- ✓ Во время деплоймента;
- ✓ Поддержка приложения:
  - Периодическое ревью процесса;
  - Периодическое тестирование на проникновения.

ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

Делится на 2 фазы:

- ✓ Пассивная фаза (сбор информации, изучение приложения);
- ✓ Активная фаза.

ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ

Активная фаза:

- Сбор информации;
- Тестирование конфигурации и Deployment Management’а;
- Тестирование ролей и прав доступа;
- Тестирование аутентификации;
- Тестирование авторизации;
- Тестирование состояния сессий;
- Тестирование входных данных;
- Обработка ошибок;
- Криптография;
- Тестирование бизнес логики;
- Тестирование клиентской части ПО.

СБОР ИНФОРМАЦИИ

Поиск утечки информации с помощью поисковиков

Цель: понять, какая конфиденциальная информация о дизайне и конфигурации приложения стала доступна непосредственно (на сайте организации) или косвенно (на стороннем веб-сайте).

Определение веб-сервера

Цель: найти версию и тип работающего веб-сервера для определения известных уязвимостей и соответствующих эксплойтов для использования во время тестирования.

```
$ nc 202.41.76.251 80
HEAD / HTTP/1.0
HTTP/1.1 200 OK
Date: Mon, 16 Jun 2003 02:53:29 GMT
Server: Apache/1.3.3 (Unix) (Red Hat/Linux)
Last-Modified: Wed, 07 Oct 1998 11:18:14 GMT
ETag: “1813-49b-361b4df6”
Accept-Ranges: bytes
Content-Length: 1179
Connection: close
Content-Type: text/html
```

```
403 HTTP/1.1 Forbidden
Date: Mon, 16 Jun 2003 02:41: 27 GMT
Server: Unknown-Webserver/1.0
Connection: close
Content-Type: text/HTML; charset=iso-8859-1
```

СБОР ИНФОРМАЦИИ

Ревью метафайлов веб-сервера

Цель: найти утечку информации о структуре приложения (файлов или папок), а также найти список файлов/папок, которые избегаются краулерами поисковых систем.


```
User-agent: *
Disallow: /cgi-bin/
Disallow: /abc/*
Disallow: */news/index/
Disallow: */office/
Disallow: */instruction/
Disallow: */remark_model/
Disallow: */remark/
Disallow: */dogovor/
Disallow: */passport/
Disallow: */settings/
```

СБОР ИНФОРМАЦИИ

Поиск приложений на веб-сервере

Цель: путем прослушивания портов, а также подстановкой названий приложений в URL, найти другие приложения, которые могут быть уязвимы.

Ревью комментариев и мета-данных

Цель: найти утечки информации в комментариях кода и мета-данных.

СБОР ИНФОРМАЦИИ

Идентификация входных точек

Цель: понять, как формируются запросы и типичные ответы от приложения.

```
GET
https://x.x.x.x/shoppingApp/buyme.asp?CUSTOMER
ID=
100&ITEM=z101a&PRICE=62.50&IP=x.x.x.x
Host: x.x.x.x
Cookie:
SESSIONID=Z29vZCBqb2IgcGFkYXdhIG15IHVzZXJuYW1l
IGlzIGZvbyBhbmQgcGFzc3dvcmQgaXMgYmFy
```

СБОР ИНФОРМАЦИИ

Идентификация входных точек

Цель: понять, как формируются запросы и типичные ответы от приложения.

```
GET
https://x.x.x.x/shoppingApp/buyme.asp?CUSTOMER
ID=
100&ITEM=z101a&PRICE=62.50&IP=x.x.x.x
Host: x.x.x.x
Cookie:
SESSIONID=Z29vZCBqb2IgcGFkYXdhIG15IHVzZXJuYW1l
IGlzIGZvbyBhbmQgcGFzc3dvcmQgaXMgYmFy
```

СБОР ИНФОРМАЦИИ

Определение web-application фреймворка

Цель: найти версию и тип работающего web-application фреймворка для определения известных уязвимостей и соответствующих эксплойтов для использования во время тестирования.

Определение архитектуры приложения

- ✓ Firewalls;
- ✓ Load balancers;
- ✓ etc.

ТЕСТИРОВАНИЕ КОНФИГУРАЦИИ

Тестирование сети/инфраструктуры

- ✓ Тестирование известных багов веб-серверов и серверов приложений;
- ✓ Тестирование административных инструментов (которые участвуют в работе приложения).

Тестирование платформы приложения

- ✓ Хранение сенситивных данных в системных файлах, логах и т.п. файлах.

ТЕСТИРОВАНИЕ КОНФИГУРАЦИИ

Тестирование содержания хранимых файлов

- ✓ Содержимое .asa и .inc файлов;
- ✓ Содержимое txt файлов, .bak, .old и других файлов на сервере.

```
/connection.inc

<?
mysql_connect(“127.0.0.1”, “root”, “”)
or die(“Could not connect”);
?>
```

ТЕСТИРОВАНИЕ КОНФИГУРАЦИИ

Тестирование HTTP методов

- ✓ PUT: метод позволяет загрузить файлы (вредоносные) на сервер;
- ✓ DELETE: метод может быть использован для удаления критически-важных для работы файлов;
- ✓ CONNECT: этот метод мог позволить клиенту использовать веб сервер как прокси;
- ✓ TRACE: этот метод отражает клиенту то, что было отравлено на сервер. Может быть использовано для атак, известных как Cross Site Scripting.

ТЕСТИРОВАНИЕ РОЛЕЙ И ПРАВ

Тестирование ролей

Цель: проверить доступность определенных компонентов и функциональностей для различных ролей в приложении.

Тестирование процесса регистрации пользователя

Цель: проверить процесс регистрации на соответствие требованиям к безопасности ПО.

ТЕСТИРОВАНИЕ АУТЕНТИФИКАЦИИ

Тестирование передачи логина/пароля

- ✓ Отправка данных используя POST запрос через HTTP протокол;
- ✓ Отправка данных используя POST запрос через HTTPS протокол;
- ✓ Отправка данных используя POST запрос через HTTPS протокол, но используя HTTP протокол;
- ✓ Отправка данных используя GET запрос через HTTPS протокол.

ТЕСТИРОВАНИЕ АУТЕНТИФИКАЦИИ

Тестирование популярных логина/пароля

- ✓ Admin/admin;
- ✓ Admin/passw0rd (или p@ssword)
- ✓ etc.

Тестирование путем подстановки URL

```
http://www.site.com/page.asp?authenticated=no

http://www.site.com/page.asp?authenticated=yes
```