Доклад

По Интернет програмиране

Тема: Често срещани уязвимости в

web приложенията

Валетин Кьосев 12а клас

1. Често срещани уязвимости в web приложенията
2. **Често срещани уязвимости в web приложенията:**

Сигурност

* Сигурността не е нещо, което се купува
* Сигурността е процес
* Сигурността е компромис
* Сигурността трябва да е основна част от дизайна на приложението
* „Тестването може да открие присъствието на грешки, но не и отсъствието им“
* Сигурността на една система е сигурността на най-слабото й място

**Общи проблеми в web приложенията**

* Невалидирани входни данни
* Грешки при контрола на достъпа
* Грешки при управлението на сесиите и идентификацията
* Cross-Site Scripting (XSS)
* Препълване на буфери
* Злонамерено вмъкване на код (code injection)
* Проблеми при обработка на грешки
* Несигурно използване на криптография
* Denial of Services
* Грешки при конфигурация

**Невалидирани входни данни**

- Обработката на входни данни е важна част от всяко приложение

* Често допускани грешки:

– Предполагане, че постъпилите данни са коректни

– Проверяване за „грешни данни“ вместо за коректни

– Не проверяване за валидност на формата на данните

– Реализиране на ненадеждни проверки

* **Частни случаи на проблема са уязвимостите от тип:**

– Cross-Site Scripting (XSS)

– Препълване на буфери

– Злонамерено вмъкване на код (code injection

* **Как да разберем, че сме уязвими?**

– Ако не проверяваме входните данни или не го правим коректно със сигурност сме уязвими

* Как да се защитим?
* – Проверявайте за ВАЛИДНИ, а не за невалидни данни!
* – Не трябва да се разчита на проверки при клиента!

Получаваните данни трябва внимателно да бъдат проверявани за:

Тип на данните (string, integer, real...)

**Сесии и идентификация**

Тъй като HTTP протоколът не поддържа сесии, разработчиците се налага да реализират собствени механизми и това води до грешки

* **Често допускани грешки**

– Използване на несигурни методи за идентификация

– „Lost password“ изисква твърде малко данни

– Пазене на повече от нужното при клиента

– Използване на предвидими сесийни идентификатори

– Съхраняване на нехеширани пароли

* **Как да разберем, че сме уязвими?**

– Като проследим всички процеси, при които се използва идентификация и сесия и проверим дали са коректни

**Как да се защитим?**

– Налагане ползването на силни пароли

– Механизъм за защита от brute force атаки на паролата

– Сигурни механизми при смяната на парола и свързаната с нея информация

– Съхранение на паролите

– Защита на предаваните данни чрез криптиране

– Обвързване на сесийният идентификатор с допълнителни данни на потребителя

– Избягвайте да давате информация за други потребители

– Не ползвайте GET при идентификация и предаване на сесийна информация

– Всеки компонент на даден сайт трябва да проверява идентификацията, а не да вярва на другите компоненти

**Препълване на буфери**

Това са проблеми в компилируемите езици, за щастие такива не се ползват често при уеб приложенията

* **Как да разберем, че сме уязвими?**

– Трябва да следим новините, свързани със сигурността на софтуера, който ползваме

– Трябва да знаем отговора на въпросите:

Ползваме ли несигурни или трудни за работа функции за обработка на низове и заделяне на памет?

Предвиждаме ли достатъчно големи буфери за входните данни?

* **Как да се защитим?**

– Използвайте функции, които по надежден начин проверяват големината на буферите и входа (strlcpy, snprintf)

– Избягвайте функции без начин за подаване на големина на буфера (gets, strcpy)

**Грешки при конфигурацията**

Конфигурацията на сървъра и на web приложението играят основна роля за сигурността

* **Често допускани грешки**

– Незакърпени проблеми в сигурността на сървърите

– Грешки в конфигурацията, позволяващи да се виждат директории и достъп до тях

– Забравени достъпни примерни файлове, архивни копия, конфигурации и т.н.

– Неправилно настроени права за достъп до файлове и директории

– Разрешени ненужни услуги, включително такива за отдалечена администрация

– Използване на стандартни акаунти с непроменени пароли

– Твърде информативни съобщения за грешки

– Неправилно конфигурирани SSL сертификати и криптографски настройки

– Използване на самоподписани сертификати

* **Как да разберем, че сме уязвими?**

– Използвайте скенери за уязвимости

– При избиране на даден сървърен софтуер се постарайте да научите какъв е препоръчителният начин сървъра да работи сигурно

– Избягвайте ползването на сървъри с лош track record

* **Как да се защитим?**

– Следете за последните проблеми в сигурността

– Инсталирайте винаги най-новите версии на софтуера и

слагайте необходимите кръпки към него

– Винаги изключвайте услугите, които не ползвате

– Редовно проверявайте за проблеми в конфигурацията,

следвайки съветите за сигурно конфигуриране

**Несигурно използване на криптография**

Повечето web приложения трябва да пазят някои важни данни в БД или във файлове и въпреки че има добри библиотеки с функции за криптиране се допускат основни грешки при реализацията

* **Често допускани грешки**

– Важни данни не се криптира

– Несигурно съхранение на ключове, сертификати и пароли

– Неправилно съхранение на данни в паметта

– Лоши източници на случайни числа

– Лош избор на алгоритъм за криптиране/хеширане

– Опити за създаване на собствен алгоритъм за криптиране/хеширане

* **Как да разберем, че сме уязвими?**

– Хората пишещи код, занимаващ се с криптиране трябва много добре да разбират какво правят

– При разглеждане на кода трябва да се обърне специално внимание на ползваните алгоритми, тяхната реализация, как защитените данни сe записват, четат и обработват

* **Как да се защитим?**

– Използвайте готови и тествани библиотеки

– Подсигурете сигурното съхранение на паролите, ключовете и сертификатитe

**Използвана литература/Източници**

<https://georgi.unixsol.org/texts/presentations/web-security.pdf>

<https://blog.superhosting.bg/most-common-website-security-threats-part2.html>

<https://technologica.com/modal/zasthita-na-prilozheniya/>