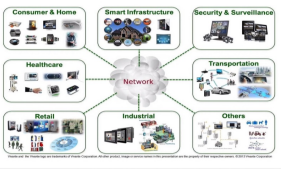
* الطبيعيين لديهم واحد بالمية من سكان العالم

لغات البرمجة



تقنيات يمكن تطبيقها وإنشاء برمجيات من خلالها واهم التقنيات لعام 2020 كالتالي:

- Blockchain
- IoT
- VR/AR
- AI
- Quantum Computing
- Real-time
- Translation
- Machine Learning
- Robotic
- Cyber Security
- Edge Computing
- SDN

الأخطاء خلال مراحل المشروع

تأثير الأخطاء البرمجية على المشروع خلال مراحل تطوير المشروع التقني

Design and architecture	Implementation	Integration testing	Customer beta test	Postproduct release
1X*	5X	10X	15X	30X

*X is a normalized unit of cost and can be expressed in terms of person-hours, dollars, etc.
Source: National Institute of Standards and Technology (NIST)†

By catching defects as early as possible in the development cycle, you can significantly reduce your development costs.

† NIST, *The Economic Impacts of Inadequate Infrastructure for Software Testing*, May 2002.

Black vs White Box Testing

ماعندي اي معلومة عن البرنامج

عندي معلومات عن البرنامج او الموقع

عندي اطلاع على الـ Source code كامل

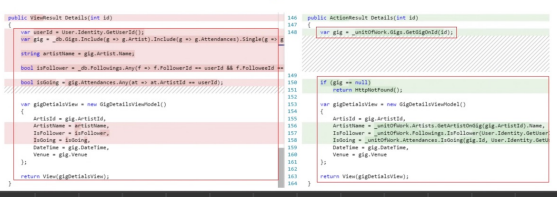
Source code اسحب فلح حلو

Microsoft Security Development Lifecycle (SDL)

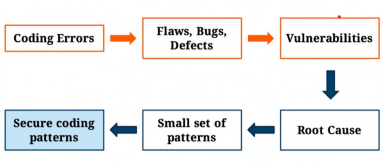
- Provide Training
- Define Security Requirements
- Define Metrics and Compliance Reporting
- Perform Threat Modeling
- Establish Design Requirements
- Define and Use Cryptography Standards
- Manage the Security Risk of Using Third-Party Components
- Use Approved Tools
- (Perform Static Analysis Security Testing (SAST
- (Perform Dynamic Analysis Security Testing (DAST
- Perform Penetration Testing
- Establish a Standard Incident Response Process

Good Coding vs Bad Coding

ماهو الفرق بين البرمجة الخاطئة والبرمجة الصحيحة ؟ 🤔



Secure Coding



البرمجة الآمنة

الهدف

- Confidentiality السرية
- Availability توفر البيانات
- Integrity السلامة

البرمجة الآمنة

المبادئ

- Minimize Attack Surfaces - تقليل فرص الهجوم والاختراقات
- Establish Secure Defaults - حماية افتراضية
- Least Privilege - أقل الصلاحيات
- Defense in Depth - الحماية العميقة
- Fail Securely
- Don't Trust Services - لا تثق في الخدمات
- Separation of Duties - تقسيم المهام
- Keep Security Simple - ابقي الحماية بشكل أسهل
- Fix Issues Correctly - اصلح المشاكل بشكل صحيح

البرمجة الآمنة

الطرق

- Authentication - التحقق
- Authorization - الصلاحيات
- Data Validation - التحقق من البيانات
- Session Management - إدارة الجلسات
- Error Handling - معالجة الأخطاء
- Logging - التعقب والمتابعة
- Encryption - التشفير

حماية قاعدة البيانات

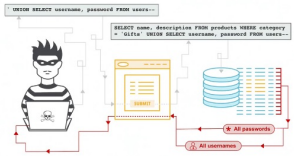
المدرب
ياسر الصيفير

CHC V2

OWASP
Open Web Application Security Project

حماية قواعد البيانات من هجوم SQL Injection

- استخدام Prepared Statements with Parameterized Queries
- استخدام تخزين الاجداثيات والتحركات Stored Procedures
- التأكد من التحقق من المدخلات
- تقليل الصلاحيات للمستخدمين وإدارتها
- حدد أنواع المدخلات
- تثبيت جدار ناري في الخوادم



https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

حماية قاعدة البيانات

المدرب
ياسر الصيفير

CHC V2

OWASP
Open Web Application Security Project

حماية قواعد البيانات

- استخدام استعلامات queries آمنة و قوية
- استخدام التحقق من صحة الإدخال وتشفير الإخراج وتأكد من معالجة الأحرف
- تأكد من كتابة المتغيرات بشكل آمن وقوي
- استخدم طريقة آمنة وحسابات محمية للدخول الى قواعد البيانات
- قم بإزالة أو تغيير كل كلمات مرور قاعدة البيانات الافتراضية. استخدم كلمات مرور / عبارات قوية أو قم بتنفيذ مصادقة متعددة العوامل
- تعطيل أي حسابات افتراضية غير مطلوبة لدعم متطلبات العمل
- يجب أن يتصل التطبيق بقاعدة البيانات بأوراق اعتماد مختلفة لكل تمييز الثقة (على سبيل المثال ، المستخدم ، المستخدم للقراءة فقط ، الضيف ، المسؤولون)

Cross Site Scripting

المدرب
ياسر الصيفير

CHC V2

OWASP
Open Web Application Security Project

حماية من Cross Site Scripting

- تشفير بيانات HTML ثم تشفير بيانات JavaScript قبل ادخال البيانات الحساسة
- تشفير بيانات JavaScript ثم تشفير بيانات HTML قبل عملية تشغيل المحتوى
- تجنب إدخال بيانات حساسة في Event Handler and JavaScript قبل التشغيل
- تشفير البيانات في HTML
- تقليل صلاحيات الدخول على properties objects
- تقليل من استخدام البيانات الحساسة فس التشغيل
- اصلاح وتحديث جميع ثغرات DOM Cross-site Scripting

https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html

تشفير البيانات

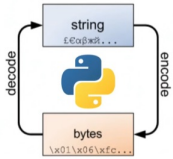
المدرب
ياسر الصيفير

CHC V2

OWASP
Open Web Application Security Project

تشفير البيانات

- استخدام خوارزميات التشفير الموثوقة والمعتمدة مثل CCM or GCM ، AES algorithm
- احفظ كلمات المرور بعد تشفيرها وإضافة salted value
- تأكد من تشفير البيانات الحساسة لعدم التسريب حتى في حالة الإختراقات
- استخدم اكثر من مفتاح تشفير
- حدد حجم البيانات المشفرة في المفتاح
- تشفير البيانات واجهات المواقع باستخدام escape syntax
- تشفير جميع البيانات وحفظها في مكان موثوق مثل الخادم
- تشفير المدخلات وقواعد البيانات
- ترميز جميع الأحرف والأرقام



https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html

التحقق من المستخدمين

المدرب ياسر العصيفير

OWASP
Open Web Application Security Project

التحقق من صحة المستخدمين

- التأكد من User IDs للمستخدم فريد
- كلمة المرور يجب أن تكون قوية ومعقدة (طول كلمة المرور، الاحرف والارقام ، الرموز)
- تفعيل طريقة آمنة لإستعادة كلمة المرور
- حفظ كلمة المرور بطريقة آمنة
- تفعيل التحقق الثنائي عبر الرسائل النصية او البريد الالكتروني
- اهم CSRF (cross-site request forgery) & Clickjacking طريقتين لطلب البرمجيات الخبيثة من المتصفح او حساب المستخدم
- تفعيل الإيقاف المؤقت للحساب في حالات الدخول المشبوهة
- تفعيل CAPTCHA للتحقق من المستخدمين

https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Utilize_Multi-Factor_Authentication

التحقق من المدخلات

المدرب ياسر العصيفير

OWASP
Open Web Application Security Project

التحقق من المدخلات

- التحقق من نوع المدخل
- تعريف انواع المدخلات المسموحة
- تحديد حد ادنى وحد اعلى للمدخلات
- استخدام Whitelisting و blacklisting
- التحقق من نصوص free-form مع Unicode
- التحقق من جهة الخوادم و من جهة واجهة المستخدمين
- منع XSS
- التحقق من الملفات المرفوعة
- التحقق من الملفات المحفوظة بحيث يكون الملف معروف مثلا png / .jpg / .gif.

https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

إدارة الجلسات

المدرب ياسر العصيفير

OWASP
Open Web Application Security Project

إدارة الجلسات Session Management

- لكل جلسة معرف فريد وخاص ID يتم انشاءه
- اسم الجلسة يجب ان يكون واضح ولا يحتوى على تفاصيل كثيرة
- طول المعرف الجلسة يكون على الاقل 128 bits (16 bytes).
- المعرف الخاص بالجلسة يجب ان يكون غير قابل للتخمين
- استخدم خادم او منصة خاصة لإدارة الجلسات والتحكم بها
- تسجيل الخروج يجب ان تنهي الجلسة وكل مايتعلق بها بالكامل
- تحديد مهلة قصيرة للاغلاق الجلسة في حالة عدم نشاطها لفترة
- إنشاء معرف جلسة جديد وإلغاء تنشيط القديم واحدة بشكل دوري
- قم بإنشاء معرف جلسة جديد إذا تغير أمان الاتصال من HTTP إلى HTTPS

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

Access Control

المدرب ياسر العصيفير

OWASP
Open Web Application Security Project

الصلاحيات

- يجب توفير سياسة خصوصية للدخول للموظفين المحللين، ادارة المشاريع، المصممين، والمطورين
- Role-Based Access Control - RBAC ←
- Discretionary Access Control DAC ← بعض وبعض
- Mandatory Access Control - MAC ← طبقات سرية للناية ماحدا يقرأه
- Permission Based Access Control

https://cheatsheetseries.owasp.org/cheatsheets/Access_Control_Cheat_Sheet.html

OWASP
Open Web Application Security Project

Privacy policy

توفير أساسيات لحماية الملفات المصدرية من ناحية التحقق من صحة بيانات المدخلات ،

والتحقق من صحة بيانات المخرجات ، إلخ.

تجهيز سياسة خصوصية ملف مكون من 9 صفحات واقل

https://www.owasp.org/index.php/Secure_Coding_Cheat_Sheet

OWASP
Open Web Application Security Project

تسجيل الإحداثيات

تساعد عملية متابعة الإحداثيات في الأنظمة والتطبيقات او المواقع على:

- متابعة جميع تحركات الدخول والخروج والاحداثيات في الخوادم والتطبيقات
- تحديد الهجمات والأحداث الأمنية
- رصد انتهاكات السياسة
- توفير معلومات حول المشاكل والظروف غير العادية
- المساعدة في الدفاع ضد الثغرات والاستغلال من خلال اكتشاف الهجوم
- المساهمة ببيانات إضافية خاصة للتحقيق في الحوادث والتي تفتقر إلى مصادر السجل الأخرى
- التدقيق ومتابعة على إضافة البيانات ، التعديل والحذف ، تصدير البيانات

https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html

تصحيح المشاكل التقنية

- استخدام Prepared Statements with Parameterized Queries
- استخدام تخزين الاحداثيات والتحركات Stored Procedures
- التأكد من التحقق من المدخلات
- تقليل الصلاحيات للمستخدمين وإدارتها
- حدد أنواع المدخلات
- تثبيت جدار ناري في الخوادم

الإختبارات الأمنية

- إزالة جميع الوظائف والملفات والتعليقات الغير الضرورية من الاكواد
- استخدم كود منظم ومحمي وموثوق افضل من انشاء كود جديد لبعض المهام
- استخدام واجهات برمجة التطبيقات (APIs) المدمجة الخاصة بالمهام لإجراء مهام نظام التشغيل.
- فحص الاكواد والتحقق منها بشكل دوري

90%

دراسة

٩٠٪ من التسريبات الأمنية حدثت بسبب الثغرات الموجودة في الملفات المصدرية (الأكواد). US Department of Homeland Security

ملخص

Authentication

- Enforce basic password security
- Implement an account lockout for failed logins
- "Forgot my password" functionality can be a problem
- For web applications, use and enforce POST method

Authorization

- Every function (page) must verify authorization to access
- Every function (page) must verify the access context
- Any client/server app must verify security on the server

Error Handling

- Don't disclose information that should remain private
- Remember to cleanup completely in an error condition

Encryption

- If storing passwords – hash with a salt value
- If you're using authentication – encrypt in transmission
- Properly seed random number generators

Data Validation

- Validate data before use in SQL Commands
- Validate data before sending back to the client
- Validate data before use in 'eval' or system commands
- Validate all data lengths before writing to buffers

Session Management

- Enforce a reasonable session lifespan
- Leverage existing session management solutions
- Force a change of session ID after a successful login

Logging

- Avoid logging sensitive data (e.g., passwords)
- Beware of logging tainted data to the logs
- Beware of logging excessive data
- Beware of potential log spoofing

Is derived from Andrew Buttner and Mark Davidson's 'Secure Code Review' class, available at http://OpenSecurityTraining.info/SecureCodeReview.html