Writing Secure ASP.NET

David Cumps – Student 2TI

cumps.david@pandora.be

http://weblogs.asp.net/CumpsD

Agenda

- Input Validatie
- Authenticatie
- System Security

Een overzicht…

- Percentage kwetsbaar : 92%
  - Cross-site scripting: 80%
  - SQL injection: 62%
  - Parameter tampering: 60%
  - Cookie poisoning: 37%
  - Database Server: 33%
  - Web Server: 23%
  - Buffer overflow: 19%

Website code zwakheden

Bron: WebCohort
http://www.imperva.com/company/news/2004-feb-02.html

Input Validatie

- Drie zeer voorkomende aanvallen
  - Cross-site scripting (XSS)
  - SQL injection aanvallen
  - Parameter Tampering
- Probleem van de hele IT industrie
  - Niet enkel met IIS/ASP/ASP.NET
- Gouden regel:

  Vertrouw je gebruikers **nooit!**

Cross-site scripting

- Invoer vertrouwen van externe bronnen
  - Form velden
  - Query string
- Die input afbeelden als output
  - Gepersonaliseerde inhoud
    (bv: Aangemelde gebruikers: user1, user2)
- Mogelijke schade
  - Hacked site
  - Diefstal of poisoning van cookie
  - …

Cross-Site Scripting

demo

Best Practice
Cross-Site Scripting

- Valideer **alle** input
  - Laat eventuele veilige tags door. (Regex ze!)
- HTML-encode input data
  - Verander “<“ in “<”, …
- ValidateRequest (ASP.NET 1.1)
  - Zet dit aan, tenzij “<“ of “>” vereist is

    Indien aan veroorzaakt dit een ASP.NET fout indien er “<“ of “>” wordt ingegeven.

Nuttig: Regular Expressions – Crash Course

http://msdn.microsoft.com/library/en-us/dnaspp/html/regexnet.asp

SQL Injection

- Invoer vertrouwen van externe bronnen
  - Form velden
  - Query string
- Input data wordt uitgevoerd
- Mogelijke schade
  - Data zichtbaar
  - Data beschadigd
  - Data vernietigd
  - …

SQL Injection

demo

Best Practice
SQL Injection

- Valideer **alle** input
  - Beperk aanvaardbare tekens
    - Getal vereist: laat enkel getallen toe
    - Gebruik regular expressions
- Gebruik altijd queries met parameters
  - Voeg nooit SQL samen met gebruiker input.
- Gebruik nooit “SA” account (admin)
  - Gebruik least privileged account

Authenticatie

- Authenticatie bepaalt wat de gebruiker kan
- Aanvallen hiertegen zijn zeer voorkomend
  - Dictionary / Brute Force aanvallen
  - Allerlei systeem aanvallen

- Hoe moeilijker info te achterhalen is hoe lager de motivatie
- Info die dient voor authenticatie moet niet (altijd) terug te halen zijn

Wachtwoorden veilig bewaren

- Bewaar een hash
  - Hash is een one-way “encryptie”
  - Collisions zijn erg onwaarschijnlijk
  - Problemen
    - 2 identieke wachtwoorden hebben dezelfde hash
    - Dictionary attack is nog steeds makkelijk
- Salted hash
  - Voeg random crypto-data toe aan het wachtwoord voor de hash digest berekend wordt
  - Dictionary attack is moeilijker

Wachtwoorden veilig bewaren

demo

Best Practice

Wachtwoorden veilig bewaren

- Gebruik salted hashes om wachtwoorden te bewaren

- Maak nooit een eigen encryptie-algoritme

Nuttig: Create Hashes - MD5, SHA1, SHA256, SHA384, SHA512
http://weblogs.asp.net/cumpsd/articles/81507.aspx

Nuttig: Encrypting the salt of a salted hash
http://www.vsdotnet.be/blogs/tommer/PermaLink,guid,66a14fe3-bfc7-4089-8b55-69480a7b78fc.aspx

System Security

- Security beperkt zich niet enkel tot één applicatie
- Controleer zelf de omgeving, admins kunnen ook zaken vergeten

IOPermissions

- Standaard Full permissions
- NTFS rechten niet voldoende
- Ingebouwde .NET security
- Least Priviliged Account

IOPermissions

demo

Best Practice
IOPermissions

- Zet een applicatie vast in zijn directory
- Beperk eventueel NTFS rechten (Deny List Folder)
- Bewaar niets confidentieel in plain text

Nuttig: Using Code Access Security with ASP .NET

http://msdn.microsoft.com/library/default.asp?url=

Onveilige DSN

- DSN is gemakzucht
- Makkelijk ‘overenthousiast’ te zijn (‘Misleidende’ GUI)

Onveilige DSN

demo

Best Practice
Onveilige DSN

- Een DSN mag enkel locatie-gegevens bevatten
  - Server
  - Database
  - Username

- DSN Password in code (**NIET** in plaintext)

Nuttig: Encrypting and decrypting data
http://www.dotnetthis.com/Articles/Crypto.htm

Besluit

- Security is niet meer weg te denken.
  - Houdt er rekening mee vanaf het begin.
  - Verdenk alles en iedereen.
  - Denk als een attacker, probeer je eigen site af te breken.

Nuttige Links

- Web Vulnerabilities Examples
  - http://www.lodoga.co.uk/attackinfo/thethreat/vulnreport.htm
- Regular Expression Library
  - http://www.regexplib.com/
- Security WebCasts
  - http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032248829&Culture=en-US
  - http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032246273&Culture=en-US
  - http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032249021&Culture=en-US

Bronnen

- Tom Mertens (Session on '*Best Practices for ASP.NET Development*')
  - http://www.vsdotnet.be/blogs/tommer/
- When Output Turns Bad: Cross-Site Scripting Explained
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncode/html/secure07152002.asp
- Improving Web Application Security: Threats and Countermeasures
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/ThreatCounter.asp
- Writing Secure Code, Second Edition
  - http://www.hcw.be/hcwcs/p.asp?p=B2735
  - http://btobsearch.barnesandnoble.com/booksearch/isbninquiry.asp?ean=9780735617223&pwb=1&displayonly=EXC

Vragen?

Never trust your users!

David Cumps – Student 2TI
cumps.david@pandora.be
http://weblogs.asp.net/CumpsD

© 2004. All rights reserved.
This presentation is for informational purposes only. There are no warranties made, express or implied, in this summary.