secunet

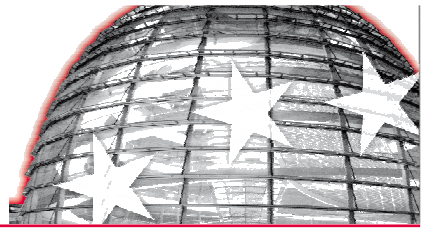
secunet Security Networks AG

Sicherheit in Web-Portalen

Hamburg, 22.11.2010

Dipl. Inform. Dirk Reimers

Vorstellung



- Dirk Reimers

  - DFN-CERT (bis 1998)

  - secunet seit 1999

  - Principal Informations-Sicherheitsmanagement,
    Secunet Security Networks AG

  - Leiter des secunet Pentest-Teams

secunet - das Unternehmen im Überblick

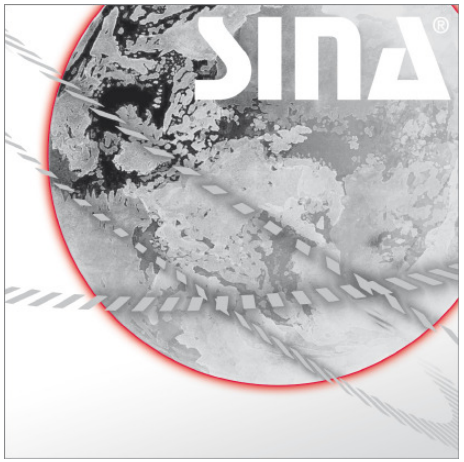
- Der führende deutsche Spezialist für komplexe IT-Sicherheitslösungen
- Sicherheitspartner der Bundesrepublik Deutschland
- Projekte in Industrie, bei Behörden und internationalen Organisationen im In- und Ausland
- Umfassende Kompetenz – kundennah
  - 4 Geschäftsbereiche
  - 7 Standorte in D, Tochterunternehmen in CH und CZ
  - 280 hoch qualifizierte Mitarbeiter

- secunet Security Networks AG
  - Gegründet 1996, börsennotiert seit 1999
  - Umsatz 2007: 41,3 Mio. Euro
  - Anteilseigner: G&D 80 %

secunet

secunet Geschäftsbereiche

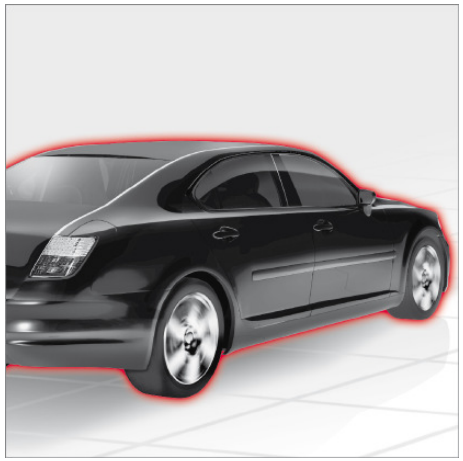
Hochsicherheit



Government

Business Security

Automotive Security



secunet

Agenda

1. Warum diese Veranstaltung?
2. Einsatzzweck von Web-Portalen
3. Klassifizierung von Schwachstellen in Web-Portalen
4. Hands-On

Einleitende Gedanken

Zu Beginn einige wichtige Grundsätze

1. Axiom: Die Welt ist schlecht

2. Axiom: Niemand verschenkt etwas

3. Axiom: Alle wollen nur den eigenen Vorteil sichern

→ Vertraue niemandem

Die Welt ist schlecht – Beispiele aus der secunet-Arbeit

- triviale Passworte
- Web-Server als offene Scheunentore
- Datenabfluss
- Kreditkarten im /LOG/ Verzeichnis eines Web-Servers
- Datenabfluss über Foren-Software

Agenda

1. **Warum diese Veranstaltung?**
2. Einsatzzweck von Web-Portalen
3. Klassifizierung von Schwachstellen in Web-Portalen
4. Hands-On

Ziele

- Einschätzung des
  - Einsatzgebietes und
  - Notwendigkeiten von Web-Portalen
- Tools zum Auffinden von Schwachstellen in Web-Portalen
- Kennenlernen von typischen Schwachstellen in Portalen
- Gegenmaßnahmen

Genug der Theorie

- Beispiel eines Angriffs mit einem gehackten Web-Server
- Angreifer hat einen Web-Server übernommen und eigene Inhalte hinterlegt
- Konkreter Angriff auf einen Benutzer per Mail
- Was kann passieren?

Agenda

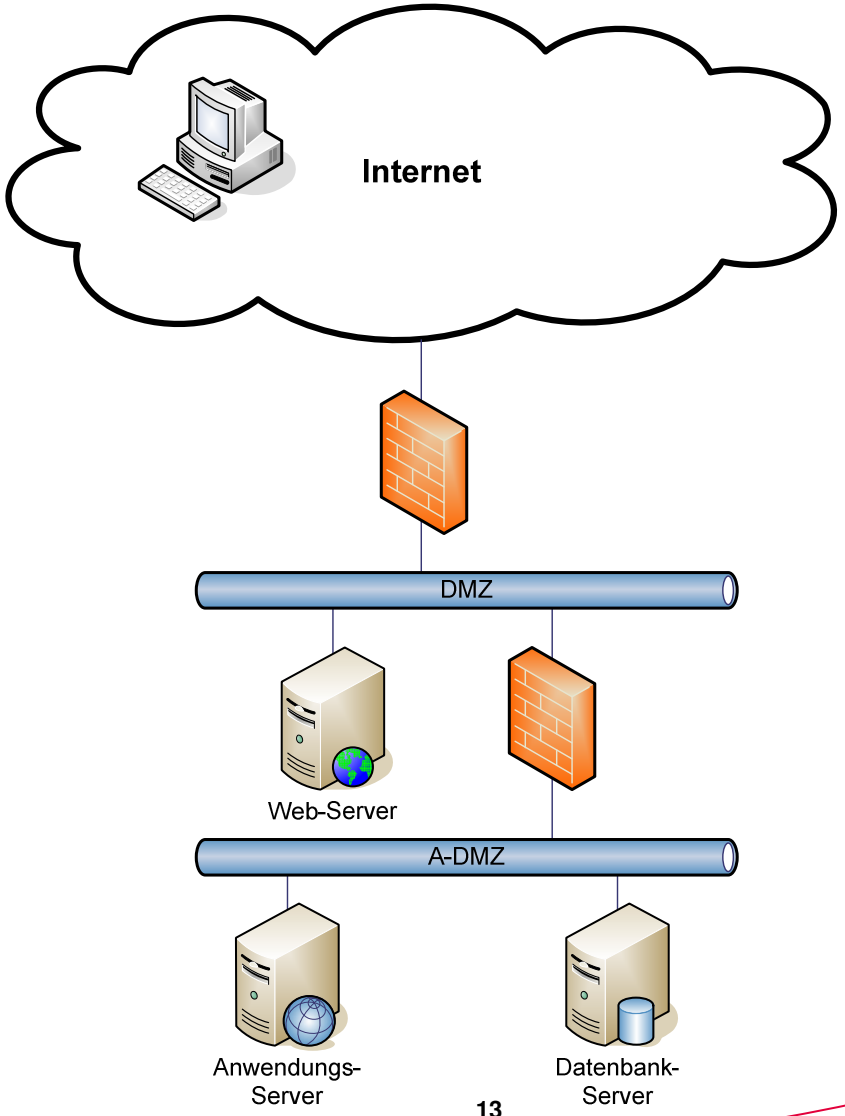
1. Warum diese Veranstaltung?
2. **Einsatzzweck von Web-Portalen**
3. Klassifizierung von Schwachstellen in Web-Portalen
4. Hands-On

Einsatzgebiete von Web-Portalen

- 1992 als die Welt noch gut war
  - Nur einige Geeks hatten Web-Seiten
  - Wenig vertraulichen gespeicherten Inhalten
  - Defacing uninteressant

- 2010
  - Portale werden B2C und B2B business critical betrieben
  - Zugriffe auf Systeme im Backend
  - geldwerte Vorteile

Struktur von Web-Portalen

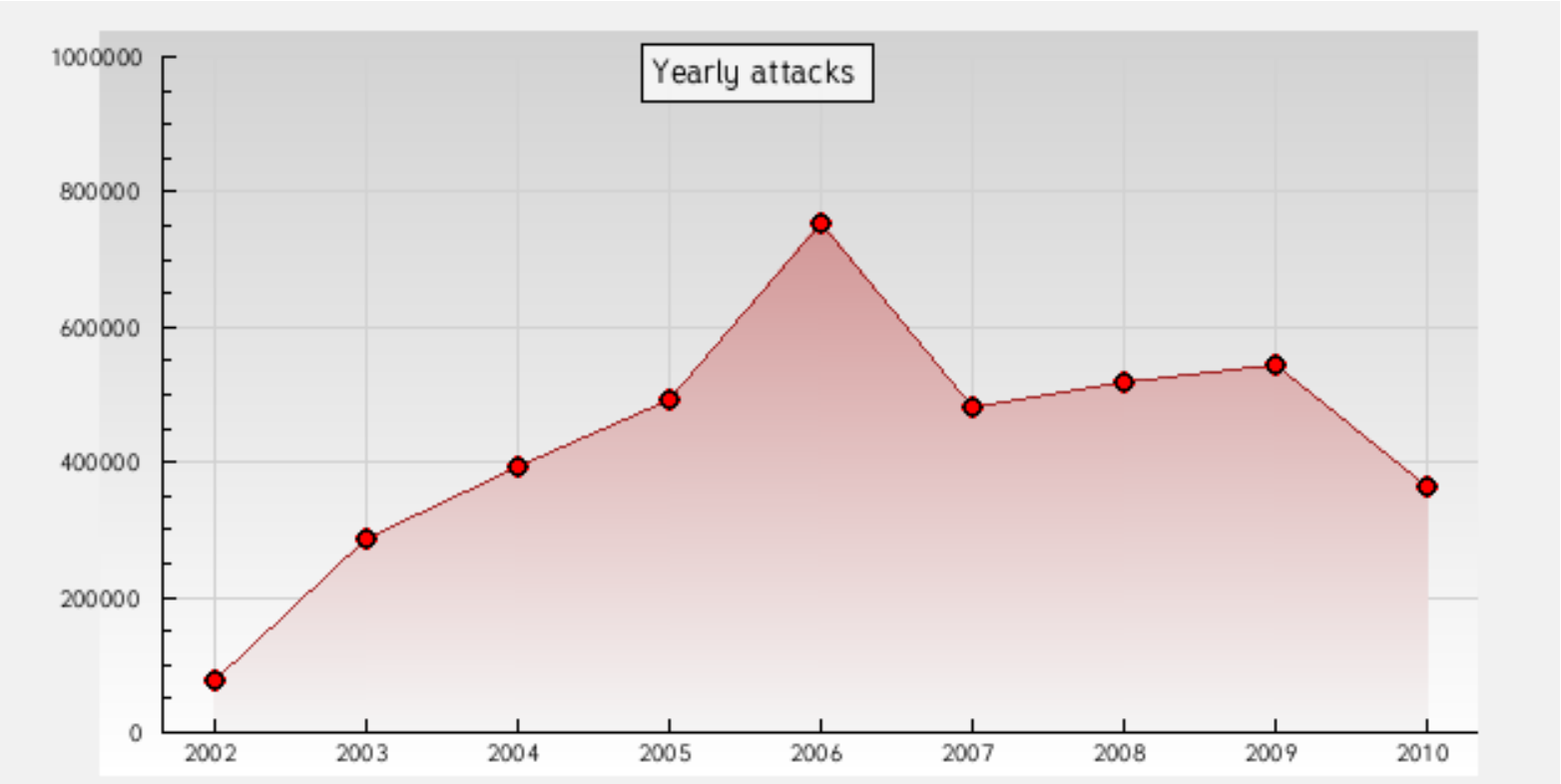


secunet

Vorteile von Web-Portal-Lösungen

- standardisierte Kommunikation
  - incl. anerkannter Sicherungsmöglichkeiten wie SSL v3 oder TLS
- geringe Anforderungen an Klienten
- unterschiedlichste Programmiersprache und Frameworks vorhanden
- Multi-TIER fähig
  - Schneller Austausch / schnelle Erweiterung einzelner Komponenten
- (geringe Bandbreitenanforderungen)

Offensichtliche Angriffe: Defacing



Quelle: http://www.zone-h.org/

secunet

übernommene Seite

Mirror saved on: 2010-06-29 17:39:02

Notified by: ir4dex
System: FreeBSD

Domain: http://watch-win.de
Web server: Apache

IP address: 82.100.220.49
Notifier stats

Ir4dex Own3d You - by gl0w

Maria Delsa dos Santos posa na porta de sua pequena casa de barro em Paulistana, no sertão do Piauí

Governo Brasileiro, o que terá de almoço e janta em sua casa hoje?

Quelle: http://www.zone-h.org/

Agenda

1. Warum diese Veranstaltung?
2. Einsatzzweck von Web-Portalen
3. **Klassifizierung von Schwachstellen in Web-Portalen**
4. Hands-On

Klassifizierung von Schwachstellen

- Schwachstellen des Betriebssystems
- Schwachstellen in den eingesetzten Anwendungen
- Schwachstellen in eigenen Entwicklungen
- Schwachstellen in der Konfiguration

Schwachstellen des Betriebssystems

- Beispiel
  - Pufferüberläufe
- Impakt
  - Hoch
    - Zugriff auf das komplette System mit administrativen Berechtigungen
    - Denial of Service
- Lösung
  - trivial wenn Patch verfügbar
  - kompliziert sonst

Schwachstellen in den eingesetzten Anwendungen

- Beispiel
  - Pufferüberlauf
  - Bypass-Directory Checking
- Impakt
  - Hoch
    - Zugriff auf das lokale Dateisystem mit den Rechten des Web-Servers
- Lösung
  - trivial wenn Patch verfügbar
  - kompliziert sonst

Schwachstellen in eigenen Entwicklungen

- Beispiel
  - Ungenügende Prüfung der Benutzereingaben
    - direkte Parameterübergabe
    - Cross-Site-Scripting
    - SQL-Injection
- Impakt
  - Hoch
    - Unberechtigter Zugriff auf Informationen
    - Änderung von Daten
- Lösung
  - Komplex je nach eingesetzter Entwicklungsumgebung

Schwachstellen in der Konfiguration

- Beispiel
  - Zugriff auf temporäre Dateien
  - Standard Benutzer
  - schlechte Passworte
- Impakt
  - Hoch
    - lesbare Passworte
    - Zugriff auf Administrationsoberflächen
- Lösung
  - trivial
    - Konfiguration anpassen

Tools zur Überprüfung von Portal-Schwachstellen

- Beschränkung auf freie Tools
- Hands on im weiteren Verlauf
- Scanner
  - w3af (w3af.sourceforge.net)
  - NeXpose Community Edition (www.rapid7.com/products/nexpose-community-edition.jsp)
  - nikto (cirt.net/nikto2)
  - wikto (www.sensepost.com/labs/tools/pentest/wikto)
  - Paros (www.parosproxy.org)
  - Burp-Suite (portswigger.net/suite/)

Tools zur Überprüfung von Portal-Schwachstellen

- Proxies
  - webscarab (www.owasp.org/index.php/Category:OWASP_WebScarab_Project)
  - ratproxy (code.google.com/p/ratproxy)
  - fiddler (www.fiddler2.com)

- Firefox Plugins
  - XSS-Me
  - SQLInject-Me
  - AccessMe
  - FoxyProxy
  - Firebug
  - Web-Developer
  - Tamper-Data
  - Cookie-Editor

Agenda

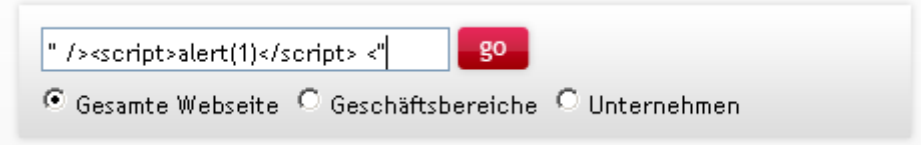
1. Warum diese Veranstaltung?
2. Einsatzzweck von Web-Portalen
3. Klassifizierung von Schwachstellen in Web-Portalen
4. **Hands-On**

Wie funktionieren Angriffe

- Fokussierung auf die Portal-Funktionen
- Angreifer senden Zeichen
- Portal wertet diese Zeichen aus
- Angriffe stecken in der Formulierung der Eingabezeichen
- Starten von ungeschützten Funktionen
- Einbetten von Skripten oder SQL Befehlen

Wie funktionieren Angriffe

- Beispiel XSS



- Suchstring:
  - " /><script>alert(1)</script> <"
- HTML
  - <input type="text" name="tx_indexedsearch[sword]" value "" /><script>alert(1)</script> <""/>
- Suchstring beendet Input Tag und startet ein Skript

Hands on

- Demonstration einiger Schwachstellen an realen Beispielen

- WAMP (Windows Apache MySQL PHP)
  - integrierte Web-Server Datenbank Lösung

- Gruyere
  - Python Anwendung mit absichtlichen Schwachstellen
  - von Google entwickelt
  - Beispielanwendung, wie man es nicht machen sollte
  - Online als Demo-System verfügbar

Hands on: einige Beispiele Live erleben

- Typische Schwachstellen in Web-Anwendungen
  - zugreifbare Verzeichnisse mit direktem Verzeichnislisting
  - Seiten mit Systeminformationen
  - alte Dateien im Web-Verzeichnis
  - triviale Passworte
  - Cookies setzen

  - Ausführen von Befehlen
  - schlechte Cookies
  - versteckte Parameter
  - Cross-Site-Scripting
  - fehlende Berechtigungsüberprüfung
  - Hochladen von Dateien
  - Cross-Site-Request-Forgery

  - SQL-Injection

zugreifbare Verzeichnisse mit direktem Verzeichnislisting

- bspw. http://www.meinserver.de/log/

- fehlerhafte Einstellungen des Web-Servers
  - Default-Einstellung für Apache
    - alle Verzeichnisse ohne index.[php|html|html] darstellen

- enthalten sonst unsichtbare Dateien/Verzeichnissen
  - Arbeitsversionen
  - „versteckte“ Dateien

- Tools finden solche Verzeichnisse
  - anhand von Kommentaren in Web-Seiten
  - per Brute Force

Seiten mit Systeminformationen

- bspw. phpinfo.php

PHP Version 5.3.0

System	Windows NT SN-HH073 5.1 build 2600 (Windows XP Professional Service Pack 3) i586
Build Date	Jun 29 2009 21:23:30
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient11\sdk,shared" "--with-enchant=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	D:\wamp\bin\apache\Apache2.2.11\bin\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,TS,VC6

- Hands on mit WAMP

secunet

alte Dateien im Web-Verzeichnis

- Editoren speichern teilweise temporäre Versionen von bearbeiteten Dokumenten
- Arbeitsversionen haben geänderte Suffixe (.php~)
- Web-Server liefern diese Dateien im Klartext aus
- Dateien können automatisiert gefunden werden
- Hands on mit WAMP

triviale Passworte

- Passworte schützen gegen unberechtigte Nutzung (auch Ihren Account)
- schlechte oder nicht vorhandene Passworte gefährden offenbar die Sicherheit
- Passworte lassen sich trivial raten
  - Brute Force
  - Datenbanken mit Standard-Passworten

Cookies

- Cookies
  - dienen zur Authentisierung
  - werden vom Host vorgegeben
- Qualität der Cookies messbar
  - statistische Analysen
- schlechte Cookie gefährden die Sicherheit der Anwendung
  - Cookies lassen sich raten
  - Cookies können vorgegeben werden
  - Angreifer können sich als Benutzer ausgeben
- Hands on mit WAMP & Gruyere

Ausführen von Befehlen

- Benutzersteuerbare Aktionen sind nicht immer an Berechtigungskonzept gebunden
- Wissen ist Macht
- Was passiert bspw. wenn man „quitserver“ testet?
- Hands on mit Gruyere

versteckte Parameter

- Parameter können in versteckten Feldern übertragen werden
- Verwendung der richtigen Parameter schafft mglw. erweiterte Rechte
- Ein Blick in den Code der Anwendung kann helfen ;-)
- http://Gruyere.appspot.com/code/?resources/editprofile.jtl
- Hands on mit Gruyere

Cross-Site-Scripting

- Auslösen von ungewollten Aktionen auf Web-Seiten
- Hinterlegen von Code in
  - Snipplets
  - Gästebüchern
  - Kommentaren
- Redefinition von Feldern
- Backend-Bomben
- Hands on mit Gruyere

Hochladen von Dateien

- Probleme durch File-Shares
- ungewollte Funktionen
  - PHP-Dateien, die auf dem Server aufgeführt werden und dort auf Ressourcen zugreifen können
- Dateien, die existierende Dateien auf dem Server ersetzen
- Hands on mit Gruyere

Cross-Site-Request-Forgery

- Voraussetzung: Gültige Sitzung zum Zielportal
- Remote Aufrufen von Funktionen des Ziel-Portals aus einer anderen Web-Seite
  - Java-Script fügt Aktionen für den Benutzer aus
- unkritisches Testobjekt
  - Logout
- kritische Funktionen:
  - Ändern des Passwortes
  - Hinzufügen von Benutzern
  - Löschen von Dateien

SQL-Injection

- Einfügen von SQL-Befehlen in Web-Seiten
  - Sonderzeichen ; (beendet einen SQL-Befehl)
  - Sonderzeichen -- (Rest der Zeile ist Kommentar)

- Danach ist die Kunst den genutzten SQL-Befehl zu raten

  - http://webserver/cgi-bin/find.cgi?ID=*42*
  - SELECT author, subjekt, text FROM artikel WHERE ID=*42*

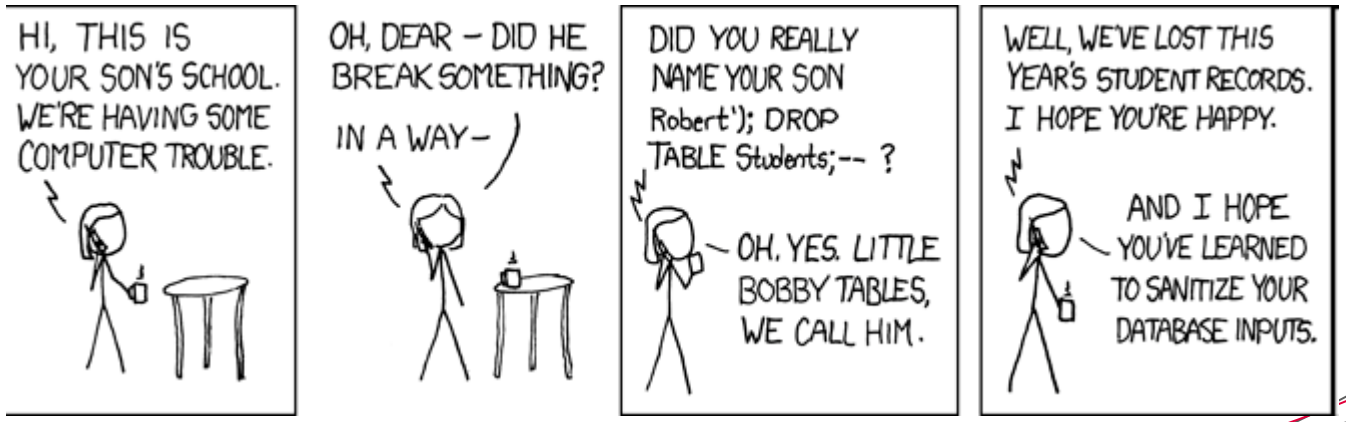
  - *http://webserver/cgi-bin/find.cgi?ID=42;UPDATE+USER+SET+TYPE="admin"+WHERE+ID=23*
  - *SELECT author, subjekt, text FROM artikel WHERE ID=42; UPDATE USER SET TYPE="admin" WHERE ID=23*

http://imgs.xkcd.com/comics/exploits_of_a_mom.png

SQL-Injection

- http://webserver/search.aspx?keyword=sql
- *SELECT url, title FROM myindex WHERE keyword LIKE '%sql%,*

- http://webserver/search.aspx?keyword=sql'+;GO+EXEC+cmdshell('format+C')+--
- *SELECT url, title FROM myindex WHERE keyword LIKE '%sql' ;GO EXEC cmdshell('format C') --%'*



http://imgs.xkcd.com/comics/exploits_of_a_mom.png

Fazit

- Die Entwicklung von Web-Portalen ist nicht trivial
- Alles wäre einfacher ohne die Anwender
- Tools testen einfache Schwachstellen
  - komplexe Sachverhalte testen am Besten die Profis

secunet Security Networks AG

Vielen Dank!

secunet Security Networks AG
Dipl. Inform. Dirk Reimers
Principal
Telefon +49 201 5454-2023
Dirk.Reimers@secunet.com