Getting a handle on SharePoint Security
(SharePoint Sicherheit im “schlüpfriegen” Griff)
Purpose today

Introduction

» Who I am ([http://archimatrix.com/jphilipp](http://archimatrix.com/jphilipp))

» 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
Existing 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

Source: [https://www.owasp.org/index.php/Research_for_SharePoint_%28MOSS%29](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 "A - G, (U), L ↔ P" security model

» **Accounts** in the Domain, organized by Domain Admins into **Global Groups**

» **Global Groups** organized by Enterprise Admins into **Universal Groups**

» **Universal Groups** or **Global Groups** organized by Resource Admins into **Local Groups** (Resource Groups)

» The Local Groups are added to ACLs and **Permissions** 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
   Federated authentication

← The Best
Applying the Model

Only easy in theory
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
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


free download and free access as authenticated users
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 it)
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, ...
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](https://github.com/alias1/sparty)
Another audit Tool

» **spscan** ([https://github.com/toddsiegel/spscan](https://github.com/toddsiegel/spscan))

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

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.

Spotting SharePoint services ...

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

**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

**Document Moved**
- 77.66.45.131
- Netgroup AVS
- 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: "b0e717fe78f5e1a: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

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
SPFisLatency: 1
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 13.0.0.4420
X-Content-Type-Options: nosniff
X-MS-InvokeApp: 1; RequireReadOnly
Date: Thu, 21 Nov ...
Cross-site scripting (XSS) vulnerability in Microsoft Office SharePoint Server 2010, Windows SharePoint Services 2.0 and 3.0 SP2, and SharePoint Foundation 2010 allows remote attackers to inject arbitrary web script or HTML via the URI, aka "SharePoint XSS Vulnerability."

Directory traversal vulnerability in Microsoft SharePoint Server 2010 SP1 and SharePoint Foundation 2010 SP1 allows remote attackers to bypass intended read restrictions for content, and hijack user accounts, via a crafted URL, aka "SharePoint Directory Traversal Vulnerability."

Cross-site scripting (XSS) vulnerability in Microsoft Office SharePoint Server 2010 Gold and SP1, and SharePoint Foundation 2010, allows remote attackers to inject arbitrary web script or HTML via the URI, aka "XSS in SharePoint Calendar Vulnerability."

Microsoft Windows SharePoint Services 3.0 SP3; SharePoint Server 2007 SP3, 2010 SP1 and SP2, and 2013
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
SPSDisco, They like to talk...
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.

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

  <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
  <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"/>
          </MoreMatches>
        </PrincipalInfo>
      </SearchPrincipalsResult>
    </SearchPrincipalsResponse>
  </soap:Body>
</soap:Envelope>
```
Nice conversation

Just be a member of a SharePoint site

» And you can:

```
POST /sites/test01/vti_bin/People.asmx HTTP/1.1
Host: _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>
```
We can talk even better together

I like WebDAV...

» And so should you
WebDAV quirks

Mapped network drives

Save As

Add a Place

- OneDrive - Personal
- Other Web Locations
- Computer
- Add a Place
Introduction to Collaborative Application Markup Language (CAML)

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.

Using SPQUERY to return SharePoint list items

Using SPQuery and CAML (Collaborative Application Markup Language) is an efficient way to retrieve data in SharePoint list. It helps us to filter and order items in the selected list. In this post, I want to introduce you to 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.

```csharp
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();
```

Source: [http://programmingshare-thienle.blogspot.com/2012/02/using-spquery-to-return-sharepoint-list.html](http://programmingshare-thienle.blogspot.com/2012/02/using-spquery-to-return-sharepoint-list.html)
**CAML Designer 2013 - ENG**

### ViewFields Order By Where Query Options

<table>
<thead>
<tr>
<th>owsFileLeafRef</th>
<th>owsMetaInfo</th>
<th>ows_ModerationStatus</th>
<th>ows_Level</th>
<th>owsLastModified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1;#123.txt.html</td>
<td>1;vti_cachedneedswritetofile;false</td>
<td>3</td>
<td>255</td>
<td>1;#2013-11-12 1:23:45</td>
</tr>
</tbody>
</table>

### CSOM Rest

```javascript
$.ajax({
  url: _spPageContextInfo.webAbsoluteUrl + '/_api/web/lists/getbytitle(documents)/items?$select=FileLeafRef&$top=10',
  type: 'GET',
  headers: { 'accept': 'application/json;odata=verbose' },
  success: function (data) {
    if (data.d.results) {
      // TODO: handle the data
      alert('Handle the data');
    }
  },
  error: function (xhr) {
    alert(xhr.statusText + ' ' + xhr.statusText);
  }
});
```

### Caml Server OM CSOM .NET Web services Powershell
Developer’s are smart!

Just do it on the client side...

» Do you like HTML5?

SPServices is a jQuery library which abstracts SharePoint's Web Services and makes them easier to use. It also includes functions which use the various Web Service operations to provide more useful (and cool) capabilities. It works entirely client side and requires no server install.
Extending SharePoint

Pitfalls
Challenges
Work Arounds
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
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>

What’s Next ...

More Security Challenges ahead
There are many more security issues to talk about:

» Office caching of secured documents (encrypted but...)

» The SharePoint App-Store challenge
   https://store.office.com/appshome.aspx?productgroup=SharePoint

» SharePoint Social: Attackers are already following you

» The crawler service “creepy crawlies” Scoping search is harder than it looks

Prototype JSOM, CAML, … quickly

Cross-Site Cascading Lists

Connect with Us – it’s free!
Fragen, Sorgen, Nöte?
Thanks for coming out!

Jan Philipp
Manager, Cyber Risk Services
Und kennt sich etwas mit Microsoft aus
jphilipp@deloitte.de
Track all changes here! (Always last Hidden Slides)

» 2013.11.10 – Jan Philipp, Alexios Fakos
   Creation of outline, timing, and frame work

» 2013.11.11 – Jan Philipp
   Adding of “Container Ship” theme

» 2013.11.11 – Alexios Fakos
   Adding of section 3 content

» 2013.11.26 – Jan Philipp
   Intro, Summary, and sections 1, 2, and 4 content added

» 2013.11.28 – Jan Philipp, Alexios Fakos
   Consistency check and dry-run. Finalized for OWASP AppSec BeNeLux release.

» 2015.01.28 – Jan Philipp
   Adapted for OWASP Frankfurt Stammtisch