Programming Windows Security

the bugs

Thanks to the folks who have taken the time to report bugs. To report a bug, please send mail to Keith Brown. I'll keep the bug list sorted in order of submission (descending by date), so you should be able to quickly scan the top to find any new additions.

Date	Chapter	Page	Description	Reported by
2003-02-03	5	150	In the first paragraph, when describing the `UOI_USER_SID` flag to `GetUserObjectInformation`, I mistakenly said that this returns the *text form* of the SID for the interactive user. This actually returns the binary SID structure.	Assarbad
2002-07-08	10	469	The last paragraph indicates that you should set CertCheckMode to a value greater than 0 to turn on CRL checking. It turns out that I got this backwards - set this to 0 to turn on CRL checking. I also made the statement that CRL checking is disabled by default. It turns out that CRL checking is actually on by default (at least in IIS 5). For some current information on CRL checking and IIS, check out knowledge base article Q258727.	Laura Granstedt
2002-01-14	6	N/A	Sam discovered even more flakiness in the "high-level" security APIs introduced in Windows NT 4. He ran into some problems using GetNamedSecurityInfo on a file-system directory on NT4 SP6a, specifically that the DACL returned apparently had its generic inherit-only ACEs converted into standard/specific permissions and merged with the other normal ACEs in the DACL. The result was that this DACL could not be reapplied back to the directory without an error. It was a mistake ever even covering the Get/Set(Named)SecurityInfo APIs. These functions, like all the rest of the functions exposed from aclapi.h are broken. I keep thinking Microsoft will address this, but it just seems that every new operating system and service pack makes them break in new and	Sam Brow

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>

interesting ways. Just before writing this errata, I tried comparing the DACLs returned from GetNamedSecurityInfo and GetFileSecurity on my "Program Files" directory on Windows XP Professional, and GetNamedSecurityInfo is returning blatantly wrong information. It's sad.

In any case, please stick to the older, more stable functions. Make sure you don't include aclapi.h anywhere in your code if you want to avoid these buggy functions.

</td>
<td></td>
</tr>
<tr>
<td>2002-01-14</td>
<td>4</td>
<td>119</td>
<td>

The code sample refers to a function, _readSecret, which is supposed to be part of the appendix. This function never made it into the book. Here's the implementation:

```
DWORD _readSecret(wchar_t* pszKey,
                  wchar_t* pszValue,
                  DWORD cchMax) {
  LSA_UNICODE_STRING usKey;
  _initString(usKey, pszKey);

  LSA_OBJECT_ATTRIBUTES oa = {sizeof oa};
  DWORD grfAccess = POLICY_GET_PRIVATE_INFORMATION;

  LSA_HANDLE hPolicy;
  NTSTATUS s = LsaOpenPolicy(0, &oa,
                             grfAccess,
                             &hPolicy);
  if (!s) {
    LSA_UNICODE_STRING* pusValue;
    s = LsaRetrievePrivateData(hPolicy,
                               &usKey,
                               &pusValue);
    if (!s) {
      if (pusValue->Length >
          ((cchMax - 1) * sizeof(wchar_t))) {
        LsaFreeMemory(pusValue);
        LsaClose(hPolicy);
        return ERROR_NOT_ENOUGH_MEMORY;
      }
      else {
        const DWORD cb = pusValue->Length;
        CopyMemory(pszValue, pusValue->Buffer, cb);
        pszValue[cb / sizeof(wchar_t)] = '\0';
      }
      LsaFreeMemory(pusValue);
    }
    LsaClose(hPolicy);
  }
  return LsaNtStatusToWinError(s);
}
```

</td>
<td>Gert-Jan Bartelds</td>
</tr>
<tr>
<td>2001-</td>
<td></td>
<td></td>
<td>

In the code sample on this page, the definition of `FOLDER_DELETE_ITEM` reads as follows:

```
#define FOLDER_DELETE_ITEM   0x00000003
```

</td>
<td>Bard</td>
</tr>
</table>

07-02	6	229	It should instead read: `#define FOLDER_DELETE_ITEM   0x00000004`	Hemmer
2001-06-18	4	94	The PVIEW.EXE tool of which I speak does not ship with the Windows 2000 resource kit; for some reason it was removed in that version. It is available on the Windows NT4 resource kit, if you have that available.	Bard Hemmer
2001-05-29	10	473	*(ed: This is more of a note than an errata. I've not personally tried this myself, but I'm happy to share Dmitry's comments with my readers...)* Keith, page 473 of Programming Windows security says: "when using Basic Authentication,web applications will have access to the client's cleartext password via a server variable named AUTH_PASSWORD. I know of no way of disabling this..." I know a way: an ISAPI filter. When using a Basic authentication a browser sends Authorization: header with Base64-encoded name/password. Here's what you do in a filter: OnPreprocHeaders you replace this header with your own that has user name only. Base64 of course. You must do that to have other useful variables populated(e.g. REMOTE_USER) OnAuthentication you add the password. The client will be happily logged in, and AUTH_PASSWORD will not show(or you can replace it with whatever message you want). I have such thing up and running.	**Dmitry Babitsky**
2001-04-19	4	108	First paragraph, last sentence: "Table A.4 in the appendix ...." should be "Table A.5 in the appendix ...."	Damian Hasse

2001-03-21	10	461	In footnote 21, the URL should have a trailing backslash. Instead of
			`http://www.cultdeadcow.com/cDc_files/cDc-351`
			please use
			`http://www.cultdeadcow.com/cDc_files/cDc-351/`
			And once again, please be forwarned that the content I'm pointing you to is pretty raw - if you're offended by vulgar language, beware.
				Friedrich Wedel

2000-	9	420	In figure 9.16, the descriptions for the security radio buttons are backwards. The top radio button should be labled	Hany

11-20			`COMAdminAccessChecksApplicationLevel`, while the bottom one should be labled`COMAdminAccessChecksComponentLevel`	Ramadan
2000-10-18	5	150	On line 5, change `GetUserObjectSecurity` to `GetUserObjectInformation`	Anonymous
2000-09-21	9	386	The footnote references Ewald (2000), but the bibliography lists this as Ewald (2001). [ed: I just received my copy of Tim's book from the publisher on March 20, 2001; congrats Tim!]	Rich Thompsen

---

Programming Windows Security: *the book*
Programming Windows Security: *the code*
Keith's blog
Pluralsight