COSEINC WINDOWS ADVISORY #1

Microsoft Agent Heap Overflow Exploit

Release Date:
December 27, 2006

Vendor:
Microsoft

Systems Affected:
Windows 2000 All Service Packs
Windows XP All Service Packs

Overview:
Microsoft Agent is a software technology that enables an enriched form of user interaction that makes learning to use a computer easier. With the software service, developers can enhance the user interface of their applications and Web pages with interactive personalities in the form of animated characters.

This feature is preinstalled on Win2k/XP and allows loading of remote character data via HTTP through Internet Explorer. Microsoft actually utilizes a custom compression algorithm to compress the character data file (.acf) which we presume is to speed up the distribution over network.

A security researcher of COSEINC Vulnerability Research Lab has discovered that Microsoft Agent has a heap overflow vulnerability. This vulnerability is triggered when Microsoft Agent parses the malformed character file in its uncompressed state in memory, by having an overly large value in a length field. This will lead to an integer overflow during the allocation of buffer. Subsequently, when data is copied to the buffer, the heap overflow will occur. The result is possible remote code execution.

Technical Details:
The vulnerability exists in the ReadWideString function in agentdpv.dll:

```
711a2cc4 mov eax,[ebp+0xc]
711a2cc7 cmp eax,ebx
711a2cc9 jz agentdpv!ReadWideStringW+0x6b (711a2d0e)
711a2ccb lea eax,[eax+eax+0x2]
711a2ccf push eax
711a2cd0 call agentdpv!operator new (711aaa6c)
```

The .acf format when uncompressed in memory, stores strings with their lengths prepended to them. To trigger the vulnerability, a large value 7FFFFFFF can be set in the length field of a string before compression takes place to create a malformed .acf file (This can be done using the Microsoft-supplied Agent Character Editor and editing the memory contents when creating the .acf file). When Microsoft Agent parses the .acf file, this length is read after uncompressing the file in memory:

```
711a2cc4 mov eax,[ebp+0xc] ; length of string
```

An integer overflow occurs presumably during the calculation of the size of the memory to allocate for a widestring using the supplied length, resulting in an allocation of 0 bytes:

711a2ccb lea eax,[eax+eax+0x2]
711a2ccf push eax
711a2cd0 call agentdpv!operator new (711aaa6c)

Sometime after, the string will be read from memory allocated earlier and copied to the buffer leading to the overflow and corrupting the heap.

711a2ce8 push ebx
711a2ce9 add edx,edx
711a2ceb push edx
711a2cec push eax
711a2ced push edi
711a2cee call dword ptr [ecx+0xc]{ole32!CMemStm::Read (771e7a1f)}

The string has been written (together with other data) to a temporary buffer as a result of the uncompressing procedure. The 2 nd DWORD in the .acf file specifies the total size of the file in its uncompressed state and is used internally to allocate the required memory for the temporary buffer.

The number of bytes to copy from this temporary buffer is apparently determined by subtracting from the total size, the size of previous data chunks and does not utilize the supplied string length.

Hence, the amount of overflow can be controlled by simply using a string of the desired length. This is why the large length of 7FFFFFFF does not result in continuous copying leading to access violation (usually in the case of an integer overflow). Consequently, an arbitrary 4-byte overwrite will occur resulting in possible code execution.

Vendor Status:
Microsoft has released a patch for this vulnerability. The patch is available at:
http://www.microsoft.com/technet/security/bulletin/ms06-068.mspx

Credit:
This vulnerability was discovered by Willow , a Windows security researcher of the COSEINC Vulnerability Research Lab (VRL).