RBS-2019-002

FileIWebControl ActiveX Control MakeSnapshot.dll Path Handling Heap Buffer Overflow

Vulnerable Program Details

Details for tested products and versions:

Vendor:	Naracontent Co,. Ltd.
Product:	FileIWebControl ActiveX Control (FileIWebControl.dll)
Version:	1.0.0.1
Component:	MakeSnapshot.dll
Component Version:	1.0.0.1

NOTE: Other versions than the one listed above are likely affected.

Credits

Carsten Eiram, Risk Based Security
Twitter: @RiskBased

Impact

The FileIWebControl ActiveX control (FileIWebControl.dll) contains a heap-based buffer overflow in a bundled library that may allow an attacker to compromise a user's system.

Vulnerability Details

The FileIWebControl ActiveX control bundles the MakeSnapshot.dll library, which exports functionality for e.g. previewing and uploading snapshots. The ActiveX control provides the bnPreviewSnapshot() and bnUploadSnapshot() methods for invoking the exported functionality.

The bnPreviewSnapshot() method accepts a single argument as defined below:

```
[id(0x00000015), helpstring("¸Þ¼µå bnPreviewSnapshot")]
void bnPreviewSnapshot([in] BSTR strPath);
```

The bnUploadSnapshot() method accepts three arguments and is defined as follows:

```
[id(0x00000017), helpstring("¸Þ¼µå bnUploadSnapshot")]
BSTR bnUploadSnapshot(
                [in] BSTR strPath,
                [in] BSTR strMainURL,
                [in] BSTR strSubURL);
```

The argument of relevance to this vulnerability is the first argument in both methods i.e. ‘strPath’, which contains the snapshot file path.

When a web page instantiates the ActiveX control and calls one of the two methods, the relevant functions in the ActiveX control eventually call the exported bnPreviewSnapshot() or bnUploadSnapshot() function in MakeSnapshot.dll. Both functions end up calling a function that checks if the ‘strPath’ argument was supplied.

```
.text:01E049E0 sub_1E049E0     proc near               ; CODE XREF: _bnPreviewSnapshot(char
*)+39p
.text:01E049E0                                         ; _bnUploadSnapshot(char *,char *,char
*)+30p
.text:01E049E0
.text:01E049E0 var_78          = dword ptr -78h
.text:01E049E0 var_74          = dword ptr -74h
.text:01E049E0 var_70          = dword ptr -70h
.text:01E049E0 ProcessInformation= _PROCESS_INFORMATION ptr -6Ch
.text:01E049E0 StartupInfo     = _STARTUPINFOA ptr -5Ch
.text:01E049E0 var_C           = dword ptr -0Ch
.text:01E049E0 var_4           = dword ptr -4
.text:01E049E0 arg_0           = dword ptr  4
.text:01E049E0
.text:01E049E0                 push    0FFFFFFFFh
.text:01E049E2                 push    offset SEH_100049E0
.text:01E049E7                 mov     eax, large fs:0
.text:01E049ED                 push    eax
.text:01E049EE                 sub     esp, 6Ch
.text:01E049F1                 push    ebp
.text:01E049F2                 push    esi
.text:01E049F3                 push    edi
.text:01E049F4                 mov     eax, ___security_cookie
.text:01E049F9                 xor     eax, esp
.text:01E049FB                 push    eax
.text:01E049FC                 lea     eax, [esp+88h+var_C]
.text:01E04A00                 mov     large fs:0, eax
.text:01E04A06                 xor     esi, esi
.text:01E04A08                 mov     [esp+88h+var_70], esi
.text:01E04A0C                 push    offset szCookieName ; Src
.text:01E04A11                 cmp     ebx, esi
.text:01E04A13                 jnz     short loc_1E04A38 ; strPath supplied?
```

If so, a 8192 byte buffer is created on the heap and zeroed.

```
.text:01E04A38 loc_1E04A38:                            ; CODE XREF: sub_1E049E0+33j
.text:01E04A38                 lea     ecx, [esp+8Ch+var_74]
.text:01E04A3C                 call    sub_1E02530
.text:01E04A41                 xor     eax, eax
.text:01E04A43                 mov     [esp+88h+var_4], esi
.text:01E04A4A                 push    2000h           ; unsigned int
.text:01E04A4F                 mov     [esp+8Ch+ProcessInformation.hProcess], eax
.text:01E04A53                 mov     [esp+8Ch+ProcessInformation.hThread], eax
.text:01E04A57                 mov     [esp+8Ch+ProcessInformation.dwProcessId], eax
.text:01E04A5B                 mov     [esp+8Ch+ProcessInformation.dwThreadId], eax
```

```
.text:01E04A5F                call    ??_U@YAPAXI@Z   ; operator new[](uint)
.text:01E04A64                push    2000h           ; size_t
.text:01E04A69                mov     ebp, eax
.text:01E04A6B                push    esi             ; int
.text:01E04A6C                push    ebp             ; void *
.text:01E04A6D                call    _memset
```

A path to the bbshot.dll library with various command line arguments is then created. The '-P' command line argument contains the input passed to the 'strPath' argument when calling one of the two ActiveX control methods.

```
.text:01E04A72                add     esp, 10h
.text:01E04A75                call    ?AfxGetModuleState@@YGPAVAFX_MODULE_STATE@@XZ ;
AfxGetModuleState(void)
.text:01E04A7A                mov     eax, [eax+4]
.text:01E04A7D                lea     edi, [esp+88h+var_78]
.text:01E04A81                mov     esi, eax
.text:01E04A83                call    sub_1E047F0
.text:01E04A88                mov     ecx, dword_1E3B21C
.text:01E04A8E                mov     edx, dword_1E3B218
.text:01E04A94                mov     eax, [eax]
.text:01E04A96                push    ebx             ; strPath
.text:01E04A97                push    ecx
.text:01E04A98                push    edx
.text:01E04A99                push    eax
.text:01E04A9A                push    offset aSBbshot_dllCDR ; "%s\bbshot.dll" -c %d -r %d -w
700 -h 10 -f gulim.ttc -P "%s"
.text:01E04A9F                push    ebp             ; char *
.text:01E04AA0                call    _sprintf        ; b0f!
```

As the string is created using a call to sprintf() without performing any boundary checks, this may lead to a heap-based buffer overflow and arbitrary code execution.

Solution

Korea Internet & Security Agency (KISA) has asked the vendor to fix the vulnerability. Currently, we are not aware of an updated version, but the vulnerable version is no longer available for download. Users are encouraged to delete the ActiveX control from their systems in the meantime.

References

RBS: RBS-2019-002[1]
VulnDB: 202056

[1] https://www.riskbasedsecurity.com/research/RBS-2019-002.pdf

Timeline

2019-01-18	Vulnerability discovered.
2019-02-01	Vulnerability reported to KrCERT/CC.
2019-04-04	Alert published to VulnDB customers.
2019-05-21	Publication of this vulnerability report.

About Risk Based Security

Risk Based Security offers clients fully integrated security solutions, combining real-time vulnerability and threat data, as well as the analytical resources to understand the implications of the data, resulting in not just security, but the right security.

Company History

Risk Based Security, Inc. (RBS) was established to support organizations with the technology to turn security data into actionable information and a competitive advantage. We do so by enhancing the research available and providing a first of its kind risk identification and evidence-based security management service.

As a data driven and vendor neutral organization, RBS is able to deliver focused security solutions that are timely, cost effective, and built to address the specific threats and vulnerabilities most relevant to the organizations we serve. We not only maintain vulnerability and data breach databases, we also use this information to inform our entire practice.

Solutions

VulnDB - Vulnerability intelligence, alerting, and third party library tracking based on the largest and most comprehensive vulnerability database in the world. Available as feature-rich SaaS portal or powerful API. Vendor evaluations including our Vulnerability Timeline and Exposure Metrics (VTEM), Cost of Ownership ratings, Code Maturity, and Social Risk Scores.

Cyber Risk Analytics - Extensive data breach database including interactive dashboards and breach analytics. Clients are able to gather and analyze security threat and data breach information on businesses, industries, geographies, and causes of loss. It also allows monitoring of domains for data breaches and leaked credentials as well as implementing a continuous vendor management program with our PreBreach data.

YourCISO - Revolutionary service that provides organizations an affordable security solution including policies, vulnerability scans, awareness material, incident response, and access to high quality information security resources and consulting services.

Vulnerability Assessments (VA) and Pentesting - Regularly scheduled VAs and pentests help an organization identify weaknesses before the bad guys do. Managing the most comprehensive VDB puts us in a unique position to offer comprehensive assessments, combining the latest in scanning technology and our own data. Detailed and actionable reports are provided in a clear and easy to understand language.

Security Development Lifecycle (SDL) - Consulting, auditing, and verification specialized in breaking code, which in turn greatly increases the security of products.