



RiskBased
SECURITY

[RBS-2019-006](#)

UNETsystem SecureSession ActiveX Control
Multiple Methods Handling Buffer Overflows

Vulnerable Program Details

Details for tested products and versions:

Vendor: UNETsystem
Product: SecureSession ActiveX Control (SecuiSFNCOMIE.dll)
Version: 4.5.5.11

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

Credits

Carsten Eiram, Risk Based Security
Twitter: @RiskBased

Impact

The SecureSession ActiveX control (SecuiSFNCOMIE.dll) contains multiple buffer overflows that may allow an attacker to compromise a user's system.

Vulnerability Details

Multiple Methods Heap Buffer Overflow

The ActiveX control provides three methods, EncipherHS2(), DecipherHS2(), and GenerateHS2(), which are defined as follows:

```
[id(0x0000001c), helpstring("method EncipherHS2")]
BSTR EncipherHS2(
    [in] BSTR bstrSource,
    [in] BSTR bstrHS,
    [in] BSTR bstrSession);

[id(0x0000001d), helpstring("method DecipherHS2")]
BSTR DecipherHS2(
    [in] BSTR bstrSource,
    [in] BSTR bstrHS,
    [in] BSTR bstrSession);

[id(0x0000001b), helpstring("method GenerateHS2")]
int GenerateHS2(
    [in] BSTR bstrEncAlgo,
```

```
[in] BSTR bstrCert,  
[in] BSTR bstrURL,  
[in] BSTR bstrSession);
```

The argument of particular interest to this vulnerability is “bstrSession”.

When either of these three methods are called, the corresponding functions in SecuiSFNCOMIE.dll eventually call a function to process the two last arguments passed to the methods. The function first checks that “http” is present in the string supplied as the “bstrHS” argument (or “bstrURL” for the GenerateHS2() method).

```
.text:10008228 ; int __stdcall sub_10008228(char *pszURL, char *pszSession, int)  
.text:10008228 sub_10008228      proc near                ; CODE XREF: M_GenerateHS2+15Cp  
.text:10008228                                     ; M_EncipherHS2+11Ap ...  
.text:10008228  
.text:10008228 var_14          = dword ptr -14h  
.text:10008228 Dst            = dword ptr -10h  
.text:10008228 var_C          = dword ptr -0Ch  
.text:10008228 var_4          = dword ptr -4  
.text:10008228 pszURL         = dword ptr 8  
.text:10008228 pszSession     = dword ptr 0Ch  
.text:10008228 arg_8          = dword ptr 10h  
.text:10008228  
.text:10008228 this = esi  
.text:10008228      mov     eax, offset loc_10010DB0  
.text:1000822D      call    _EH_prolog  
.text:10008232      push   ecx  
.text:10008233      push   ecx  
.text:10008234      push   ebx  
.text:10008235      push   this  
.text:10008236      push   edi  
.text:10008237      push   offset aHttp      ; "http"  
.text:1000823C      push   [ebp+pszURL]      ; Str  
.text:1000823F      mov    this, ecx  
.text:10008241      call  ds:strcmp  
.text:10008247      pop    ecx  
.text:10008248      test   eax, eax  
.text:1000824A      pop    ecx  
.text:1000824B      jz     @return_1         ; "http" not found in string?
```

The length of the URL is then checked to ensure that it’s not longer than 1024 characters.

```
.text:10008251      push   [ebp+pszURL]      ; Str  
.text:10008254      call  strlen  
.text:10008259      mov    ebx, 400h  
.text:1000825E      pop    ecx  
.text:1000825F      cmp    eax, ebx  
.text:10008261      ja     @return_1
```

If the check succeeds, the supplied “bstrHS” / “bstrURL” and “bstrSession” arguments are copied to an object on the heap via calls to strcpy().

```
.text:10008267          mov     edi, [ebp+pszSession]
.text:1000826A          lea    eax, [this+444h]
.text:10008270          push   edi             ; Source
.text:10008271          push   eax             ; Dest
.text:10008272          call   strcpy          ; b0f!
.text:10008277          push   [ebp+pszURL]    ; Source
.text:1000827A          lea    eax, [this+44h]
.text:1000827D          push   eax             ; Dest
.text:1000827E          call   strcpy
```

As no bounds checks were performed for the “bstrSession” argument, this may lead to a heap-based buffer overflow.

DestroySession2() Method Stack Buffer Overflow

The DestroySession2() method accepts a single argument and is defined as follows:

```
[id(0x0000000d), helpstring("method DestroySession2")]
int DestroySession2([in] BSTR bstrHSURL);
```

When the method is called, the corresponding function in SecuiSFNCOMIE.dll performs various inconsequential processing before eventually copying the supplied “bstrHSURL” argument to a 1024 byte stack buffer via a strcpy() call.

```
.text:1000A44B          push   ebx             ; int
.text:1000A44C          lea    ecx, [ebp+var_14] ; this
.text:1000A44F          call   ?GetBuffer@CString@@QAEPADH@Z ; CString::GetBuffer(int)
.text:1000A454          push   eax             ; Source
.text:1000A455          lea    eax, [ebp+szDest] ; char[1024]
.text:1000A45B          push   eax             ; Dest
.text:1000A45C          call   strcpy
```

As no bounds checks are performed this may lead to a stack-based buffer overflow.

CheckExistKey() Method Stack Buffer Overflow

The CheckExistKey() method accepts a single argument and is defined as follows:

```
[id(0x0000000c), helpstring("method CheckExistKey")]
int CheckExistKey([in] BSTR bstrHSURL);
```

When the method is called, the corresponding function in SecuiSFNCOMIE.dll performs various inconsequential processing before eventually copying the supplied “bstrHSURL” argument to a 1024 byte stack buffer via a strcpy() call.

```
.text:1000A136          push   ebx             ; int
.text:1000A137          lea    ecx, [ebp+oCString] ; this
.text:1000A13A          call   ?GetBuffer@CString@@QAEPADH@Z ; CString::GetBuffer(int)
```

```
.text:1000A13F          push    eax                ; Source
.text:1000A140          lea    eax, [ebp+szDest] ; char[1024]
.text:1000A146          push    eax                ; Dest
.text:1000A147          call   strcpy
```

As no bounds checks are performed this may lead to a stack-based buffer overflow.

Solution

The vendor has deprecated the ActiveX control, and KrCERT/CC plans to set the kill-bit.

References

RBS: RBS-2019-006¹
VulnDB: 202018, 202019, 202020

Timeline

2019-01-28	Vulnerabilities discovered.
2019-02-01	Vulnerabilities reported to KrCERT/CC.
2019-04-04	Alerts published to VulnDB customers.
2019-05-21	Publication of this vulnerability report.

¹ <https://www.riskbasedsecurity.com/research/RBS-2019-006.pdf>

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.