RBS-2014-005

3S Pocketnet Tech MediaViewer ActiveX Control
Multiple Buffer Overflow Vulnerabilities
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>About Risk Based Security</td>
<td>3</td>
</tr>
<tr>
<td>Company History</td>
<td>3</td>
</tr>
<tr>
<td>Solutions</td>
<td>3</td>
</tr>
<tr>
<td>Vulnerable Program Details</td>
<td>4</td>
</tr>
<tr>
<td>References</td>
<td>4</td>
</tr>
<tr>
<td>Credits</td>
<td>4</td>
</tr>
<tr>
<td>Vulnerability Details</td>
<td>5</td>
</tr>
<tr>
<td>SetDisplayText() Method</td>
<td>5</td>
</tr>
<tr>
<td>Multiple Methods sONVIF_URL Argument</td>
<td>6</td>
</tr>
<tr>
<td>GetONVIFStreamUri() Method</td>
<td>8</td>
</tr>
<tr>
<td>SaveCurrentImage() Method</td>
<td>9</td>
</tr>
<tr>
<td>SaveCurrentImageEx() Method</td>
<td>11</td>
</tr>
<tr>
<td>StartRecord(), StartRecordEx(), and StartScheduledRecord() Methods</td>
<td>12</td>
</tr>
<tr>
<td>Solution</td>
<td>14</td>
</tr>
<tr>
<td>Timeline</td>
<td>14</td>
</tr>
</tbody>
</table>
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 in early 2011 to better support the many users and initiatives of the Open Security Foundation - including the OSVDB and DataLossDB projects. RBS was created to transform this wealth of security data into actionable information 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. Available as feature-rich SaaS portal or powerful API

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.

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.
Vulnerable Program Details

Details for the tested product and version:

Vendor: 3S Pocketnet Tech
Product: Speed Dome N4011 Network Camera
Version: 1.04

Vendor: 3S Pocketnet Tech
Product: 3S VMS (PC NVR)
Version: 1.2

Component: MediaViewer ActiveX Control
(MediaClientAxCtrl.dll / PocketNetMediaClientAxCtrl.dll)
Version: 1.0.23.2, 1.1.2.0, and 1.2.4

NOTE: Other products and versions are very likely also affected.

References

RBS: RBS-2014-005
OSVDB / VulnDB: 115471¹, 115472², 115473³, 115474⁴, 115475⁵, 115476⁶
CVE: CVE-2014-9263
ZDI: ZDI-14-393, ZDI-14-394, ZDI-14-395, ZDI-14-396, ZDI-14-397

Credits

Carsten Eiram, Risk Based Security

Twitter: @CarstenEiram
Twitter: @RiskBased

¹ http://www.osvdb.org/show/osvdb/115471
² http://www.osvdb.org/show/osvdb/115472
³ http://www.osvdb.org/show/osvdb/115473
⁴ http://www.osvdb.org/show/osvdb/115474
⁵ http://www.osvdb.org/show/osvdb/115475
⁶ http://www.osvdb.org/show/osvdb/115476
Vulnerability Details

IP cameras and other products in the 3S product line provide a web-based interface for e.g. configuring device settings or viewing feeds. When accessed for the first time using Internet Explorer, an ActiveX control is installed on the client system. This bundled ActiveX control has been determined to be affected by multiple critical vulnerabilities, allowing attackers to compromise client systems when viewing a malicious web page.

SetDisplayText() Method

One of the methods supported by the ActiveX control is `setDisplayText()`, which is defined as:

```c
VARIANT_BOOL SetDisplayText(
    [in] BSTR szText,
    [in] long R,
    [in] long G,
    [in] long B);
```

When the method is called, the handling function immediately calls another function with the supplied arguments.
This function then copies the supplied string via the ‘szText’ argument into a 520 wide-character buffer at offset 684h of an object stored on the heap.

```
; Attributes: bp-based frame

; int __stdcall SetDisplayText(wchar_t *, int, int, int)
SetDisplayText proc near

arg_0= dword ptr 8
arg_4= dword ptr 0Ch
arg_8= dword ptr 10h
arg_C= dword ptr 14h

push    ebp
mov     ebp, esp
push    esi
push    [ebp+arg_0]   ; wchar_t *
mov     esi, ecx
lea     eax, [esi+684h]
push    eax            ; wchar_t *
call    wcscpy         ; Buffer overflow
```

The copy operation uses wcscpy() without performing any bounds checks, leading to a heap-based buffer overflow and control of the program flow.

**Multiple Methods sONVIF_URL Argument**

Three methods supported by the ActiveX control are GetONVIFDeviceInformation(), GetONVIFProfiles(), and GetONVIFStreamUri(), which are defined as:

```
VARIANT_BOOL GetONVIFDeviceInformation([in] BSTR sONVIF_URL);

VARIANT_BOOL GetONVIFProfiles([in] BSTR sONVIF_URL);

VARIANT_BOOL GetONVIFStreamUri(
    [in] BSTR sONVIF_URL,
    [in] BSTR sProfileToken);
```

This analysis only covers the GetONVIFProfiles() method, but the other methods are similarly affected.
When the function handling the method is called, it checks that the 'sONVIF_URL' argument was supplied, and then converts it before passing the string as argument to another function.

This function allocates memory for a 5640 byte structure on the heap and copies the argument into offset 8 using `strcpy()` without performing any bounds checks. This allows triggering a heap-based buffer overflow and gaining control of the program flow.
**GetONVIFStreamUri() Method**

One of the methods supported by the ActiveX control is `GetONVIFStreamUri()`, which is defined as:

```c
VARIANT_BOOL GetONVIFStreamUri(
    [in] BSTR sONVIF_URL,
    [in] BSTR sProfileToken);
```

When the method is called, it starts out by converting the supplied arguments after which another function is called.
This function creates a 5640 byte structure on the heap and proceeds to copy both arguments into it using `strcpy()` without performing any bounds checks. This leads to heap-based buffer overflows and allows gaining control of the program flow.

Please note that the problem with the first argument, `'SONVIF_URL'`, is already covered by the previous issue, so this issue only pertains to the copying of the `'sProfileToken'` argument into offset 408h.

**SaveCurrentImage() Method**

One of the methods supported by the ActiveX control is `SaveCurrentImage()`, which is defined as:

```c
VARIANT_BOOL SaveCurrentImage( 
    [in] long format, 
    [in] BSTR filename); 
```

When the method is called, it checks that the `filename` argument was supplied and, if so, copies it into a 1040 wide-character buffer on the stack.
As the 'filename' argument is copied straight into the buffer using `wcsncpy()` without performing any bounds checks, this allows triggering a stack-based buffer overflow and gaining control of the program flow.
SaveCurrentImageEx() Method

One of the methods supported by the ActiveX control is `SaveCurrentImageEx()`, which is defined as:

```c
VARIANT_BOOL SaveCurrentImageEx(  
    [in] long format,  
    [in] BSTR filename,  
    [in] long sec,  
    [in] long usec);
```

When the method is called, it checks that the ‘filename’ argument was supplied and, if so, copies it into a 260 wide-character buffer on the stack.
As the 'filename' argument is copied straight into the buffer using wcsncpy() without performing any bounds checks, this allows triggering a stack-based buffer overflow and gaining control of the program flow.

**StartRecord(), StartRecordEx(), and StartScheduledRecord() Methods**

Three methods supported by the ActiveX control are StartScheduledRecord(), StartRecord(), and StartRecordEx(), which are defined as:

VARIANT_BOOL StartRecord([in] BSTR fn);

VARIANT_BOOL StartRecordEx(
    [in] BSTR fn,
    [in] long sec,
    [in] long usec);

VARIANT_BOOL StartScheduledRecord(
    [in] BSTR fn,
    [in] long weekdays,
    [in] long startHour,
    [in] long startMin,
    [in] long duration);

This analysis is of the StartRecord() method, but the other functions are similarly affected. When the function handling the method is called, it immediately calls another function.
This function in turn retrieves a virtual function pointer and jumps to it.

The called function copies the supplied string argument straight into offset CB8h of an object on the heap using `wcscpy()` without performing any bounds checks.

This allows triggering a heap-based buffer overflow and gaining control of the program flow.
Solution

The vendor was not responsive and no fixes have been made available. Set the kill-bit for the vulnerable ActiveX control.

Timeline

2014/05/13  Vulnerabilities discovered.
2014/05/13  Vulnerabilities reported to Zero Day Initiative (ZDI)
2014/07/02  Case contracted with ZDI.
2014/08/13  ZDI contacts the vendor (1st attempt)
2014/09/04  ZDI contacts the vendor (2nd attempt)
2014/10/13  ZDI contacts the vendor (3rd attempt)
2014/11/05  ZDI contacts ICS-CERT (1st attempt)
2014/11/21  ZDI contacts ICS-CERT (2nd attempt)
2014/12/04  ZDI publishes advisories. OSVDB entries published and details made available to VulnDB customers.7
2015/05/06  Publication of this vulnerability report.