RBS-2015-001

EverFocus Electronics Corp
EPlusOcx ActiveX Control Multiple Stack Buffer Overflows
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>LiveStream() Method Handling Stack Buffer Overflow</td>
<td>5</td>
</tr>
<tr>
<td>SendHttpRequest() Method Handling Stack Buffer Overflow</td>
<td>6</td>
</tr>
<tr>
<td>OpenArchive() Function Stack Buffer Overflow</td>
<td>8</td>
</tr>
<tr>
<td>Solution</td>
<td>10</td>
</tr>
<tr>
<td>Timeline</td>
<td>10</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 tested products and versions:

Vendor: EverFocus Electronics Corporation
Product: ECOR264-8D2
Version: 1.0.13

Vendor: EverFocus Electronics Corporation
Product: ECOR960-16X1
Version: 1.5.26

Component: EPlusOcx ActiveX Control (ePlusOcx.ocx)
Version: 1.0.6.24

NOTE: It is suspected that many if not all DVRs in the EverFocus ECOR DVR product line bundle the vulnerable ActiveX control. Other models and versions are, therefore, likely affected.

References

RBS: RBS-2015-001
OSVDB / VulnDB: 116600\(^1\), 116601\(^2\), and 116602\(^3\)
CVE: N/A

Credits

Carsten Eiram, Risk Based Security
Twitter: @CarstenEiram
Twitter: @RiskBased

\(^1\) [http://www.osvdb.org/show/osvdb/116600](http://www.osvdb.org/show/osvdb/116600)
\(^2\) [http://www.osvdb.org/show/osvdb/116601](http://www.osvdb.org/show/osvdb/116601)
\(^3\) [http://www.osvdb.org/show/osvdb/116602](http://www.osvdb.org/show/osvdb/116602)
Vulnerability Details

DVRs in the EverFocus ECOR product line provide a web-based interface for e.g. configuring device settings. When accessed for the first time using Internet Explorer, an ActiveX control, ePlusOcx.ocx, is installed on the client system. This bundled ActiveX control has been determined to be affected by three critical vulnerabilities, all allowing attackers to compromise the client system when viewing a malicious web page.

**LiveStream() Method Handling Stack Buffer Overflow**

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

```c
long LiveStream(
    BSTR IP,
    long Port,
    BSTR SessionID,
    long Mask);
```

When the method is called, a debug string is crafted based on the supplied arguments.

```
.text:10017A60 M_LiveStream proc near ; DATA XREF: .rdata:10185EC0
.text:10017A60 szOutputString = byte ptr -208h ; char[260]
.text:10017A60 szBuf = byte ptr -104h ; char[256]
.text:10017A60 var_4 = dword ptr -4
.text:10017A60 pwzIP = dword ptr 4
.text:10017A60 lPort = dword ptr 8
.text:10017A60 pwzSessionID = dword ptr 0Ch
.text:10017A60 lMask = dword ptr 10h
.text:10017A60 .text:10017A60 sub esp, 208h
.text:10017A60 mov eax, dword_101DCD64
.text:10017A60 xor eax, esp
.text:10017A6D mov [esp+208h+var_4], eax
.text:10017A74 push ebx
.text:10017A75 mov ebx, [esp+20Ch+pwzIP]
.text:10017A7C push ebp
.text:10017A7D mov ebp, [esp+210h+pwzSessionID]
.text:10017A84 push esi
.text:10017A85 push edi
.text:10017A86 push 103h ; size_t
.text:10017A8B lea eax, [esp+21Ch+szOutputString+1] ; char[260]
.text:10017A8F push 0 ; int
.text:10017A91 push eax ; void *
.text:10017A92 mov esi, ecx
.text:10017A94 mov [esp+224h+szOutputString], 0 ; char[260]
```
The string is written to a 260 byte stack buffer using `sprintf()` without performing any bounds checks. By providing sufficiently long arguments to the method, it is possible to trigger a stack-based buffer overflow and gain control of the program flow.

### SendHttpRequest() Method Handling Stack Buffer Overflow

The ActiveX control supports the `SendHttpRequest()` method, which takes eight arguments:

```c
long SendHttpRequest(
    long nVerb,
    BSTR szID,
    BSTR szIp,
    long nPort,
    BSTR szUsr,
    BSTR szPw,
    BSTR szObject,
    BSTR szXmlCmd);
```

When the method is called, a debug string is crafted based on the supplied arguments.
The string is written to a 256 byte stack buffer using \texttt{sprintf()} without performing any bounds checks. By providing sufficiently long arguments to the method, it is possible to trigger a stack-based buffer overflow and gain control of the program flow.
OpenArchive() Function Stack Buffer Overflow

The ActiveX control supports the five following methods used for handling archives:

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>long ArchiveStream(</code></td>
<td><code>BSTR IP,</code> long Port, <code>BSTR SessionID,</code> <code>unsigned long Mask,</code> <code>long StartTime,</code> <code>long EndTime,</code> <code>long MaxFileLenth)</code></td>
</tr>
<tr>
<td><code>long ArchiveStreamEx(</code></td>
<td><code>BSTR IP,</code> long Port, <code>BSTR SessionID,</code> <code>unsigned long Mask,</code> <code>long StartTime,</code> <code>long EndTime,</code> <code>long MaxFileLenth,</code> <code>short SelectFile,</code> <code>BSTR FileName)</code></td>
</tr>
<tr>
<td><code>long ArchiveStreamExRV(</code></td>
<td><code>BSTR IP,</code> long Port, <code>BSTR SessionID,</code> <code>unsigned long Mask,</code> <code>long StartTime,</code> <code>long EndTime,</code> <code>long VideoType,</code> <code>long MaxFileLength,</code> <code>short SelectFile,</code> <code>BSTR FileName)</code></td>
</tr>
<tr>
<td><code>long ArchiveStreamExNRV(</code></td>
<td><code>BSTR IP,</code> long Port, <code>BSTR SessionID,</code> <code>unsigned long Mask,</code> <code>long StartTime,</code> <code>long EndTime,</code> <code>long ArchiveType,</code> <code>long VideoType,</code> <code>long MaxFileLength,</code> <code>short SelectFile,</code> <code>BSTR FileName)</code></td>
</tr>
<tr>
<td><code>long ArchiveStreamExPlus(</code></td>
<td><code>BSTR IP,</code> long Port, <code>BSTR SessionID,</code> <code>double Mask,</code> <code>long StartTime,</code> <code>long EndTime,</code> <code>long FileSize,</code> <code>short SelectFile,</code> <code>BSTR FileName)</code></td>
</tr>
</tbody>
</table>
When called, each method may eventually end up calling the `OpenArchive()` function within `ePlusOcx.ocx` to retrieve an archive from a device at a specified IP.

The `OpenArchive()` function starts out by creating a debug string ("OpenArchive ip=%s max=%d\n") using the ‘IP’ string argument supplied to each of the five methods.

The string is written to a 256 byte stack buffer using `sprintf()` without performing any bounds checks. By providing sufficiently long ‘IP’ argument to one of the five methods, it is possible to trigger a stack-based buffer overflow and gain control of the program flow.
Solution

The vendor was not responsive, and we are not aware of any available fix for these vulnerabilities.

Timeline

2014/10/23 Vulnerabilities discovered.
2014/12/08 Emailed vendor to obtain details for security contact.
2015/01/01 OSVDB entry published and details made available on VulnDB\(^4\).
2015/05/21 Publication of this vulnerability report.

\(^4\) https://www.riskbasedsecurity.com/risk-data-analytics/vulnerability-database/