doi 10.3969/j.issn.1671-1122.2014.10.006

典型 Adobe Flash Player 漏洞简介与原理分析

孟正，曾天宁，马洋洋，文伟平

（北京大学软件与微电子学院，北京 102600）

摘　要：随着 Flash 文件在网络上的应用日益广泛，Adobe Flash Player 的安全问题受到人们越来越多的关注，它的每一个漏洞都有引发巨大安全问题的可能性。文章首先从 Flash 客户端技术和 Flash 三维可视化分析两个方面对 Adobe Flash Player 的运行机制进行了介绍，研究了 ActionScript 语言、Flash 渲染模型、事件机制、Flash 三维图形显示、Stage3D 硬件加速和 Stage3D 三维建模的特性；接下来描述了 SWF 文件的格式信息，对 SWF 文件的文件头和标签结构进行介绍；然后结合 CVE 网站的统计信息，对 Adobe Flash Player 漏洞进行分类，将其划分为 Flash 文件格式漏洞、Flash 拒绝服务漏洞、Flash 跨站脚本攻击漏洞和 Flash 欺骗攻击漏洞等 4 大类；随后对漏洞分析技术进行了详细介绍，建立了针对 Adobe Flash Player 的漏洞分析技术模型；最后以 10 个典型的 Adobe Flash Player 漏洞作为实例，经过信息收集、数据流跟踪和漏洞原理分析等过程，得到了漏洞的产生机理。

关键词：Adobe Flash Player；漏洞分类；漏洞分析；SWF 文件格式

中图分类号：TP309　　文献标识码：A　　文章编号：1671-1122（2014）10-0031-07

Introduction and Analysis of Adobe Flash Player Vulnerabilities

MENG Zheng, ZENG Tian-ning, MA Yang-yang, WEN Wei-ping

(*School of Software&Microelectronics, Peking University, Beijing* 102600*, China*)

Abstract: As the application of Flash file in the network is becoming more and more wide, the security problems of Adobe Flash Player have also attracted more and more attentions. Every vulnerability has a possibility to arise serious security problem. This dissertation first describes the operation mechanism of Adobe Flash Player from the two aspects of Flash client technology and Flash 3D visualization analysis, gives a research on the characters of ActionScript language, Flash rendering model, event mechanism, Flash three dimensional graphic display, Stage3D hardware speeding and Stage3D modeling. Then the format of SWF file is described, and the file heading and the label structure are introduced. Combining with the statistic information of CVE website, the article takes a classification on the vulnerabilities of Adobe Flash Player. These four types of vulnerabilities are Flash file format vulnerability, Flash service denial vulnerability, Flash cross site scripting vulnerability and Flash spoofing attack vulnerability. Then the vulnerabilities classification method and the vulnerabilities analysis technology of Adobe Flash Player are described in detail and the technical model for vulnerability analysis targeting on Adobe Flash Player is built up. At last, ten typical vulnerabilities in Adobe Flash Player are taken as the practical examples. After the processes of information collection, data flow tracking and vulnerability principle analysis, the vulnerability production mechanism is drawn out finally.

Key words: Adobe Flash Player; vulnerabilities classification; vulnerabilities analysis; SWF file format

收稿日期：2014-06-06

基金项目：国家自然科学基金 [61170282]

作者简介：孟正（1990-），男，河北，硕士研究生，主要研究方向：漏洞分析和漏洞挖掘；曾天宁（1990-），男，山东，硕士研究生，主要研究方向：网络与系统安全；马洋洋（1989-），男，山东，硕士研究生，主要研究方向：网络与系统安全；文伟平（1976-），男，湖南，副教授，博士，主要研究方向：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等。

0 引言

安全漏洞作为网络中一种重要的潜在威胁，引发了许多安全问题，包括针对这些漏洞的木马、蠕虫以及病毒等。由 2008—2013 年的 Cert 报告可知，由安全漏洞引起的攻击数量占据了安全事件总数的 50% 以上。

Flash Player 是一款组件，它用于增强 Web 浏览，也允许用户对各种多媒体 Web 内容进行查看。Adobe Flash Player 是一个被广泛使用的播放软件，它的每一个漏洞都有引发巨大安全问题的可能性。随着 Flash 文件在网络上的应用日益广泛，Adobe Flash Player 的安全问题受到了人们越来越多的关注。在网页上加载由黑客精心构造的 flash 畸形文件后，使用浏览器插件 flash.ocx 将其下载下来，然后通过 Adobe Flash Player 打开 flash 文件，就有可能触发漏洞，从而造成权限泄露或系统崩溃等 [1,2]。有些 Flash 插件漏洞会被应用于浏览器，除了对 IE 浏览器本身产生影响，还会影响其他第三方浏览器，甚至会影响到全部和 Flash 有关的应用。攻击者可以对特制的 Flash 动画（SWF）文件进行构建，从而利用这些 Flash 漏洞。一旦用户对含有特制 SWF 文件的网站进行访问，则有可能允许远程代码执行。有些特制的 SWF 文件会作为电子邮件附件被发送，在这种情况下，只有当此电子邮件被打开时，用户才会面临风险 [3]。

近年来，国内外科研工作者在 Adobe Flash Player 的漏洞分析方面展开了一些研究。文献 [1] 分析了 Flash 应用程序漏洞及其利用方法，但针对的 Adobe Flash Player 的版本是 7.0-9.0，不具有代表性，另外该文完全是从 SWF 文件格式处理的角度进行阐述的，不能反映出 Adobe Flash Player 漏洞的一般特征。文献 [2] 仅对 Adobe Flash Player 的整数溢出漏洞原理进行简单描述，没有深入挖掘该类漏洞的本质特点。

为弥补上述缺陷，本文首先分析 Adobe Flash Player 的运行机制，描述 SWF 文件的格式信息，然后详细介绍 Adobe Flash Player 漏洞的分类信息和分析方法，并以 10 个典型的 Adobe Flash Player 漏洞作为实例，详细分析漏洞的产生机理。

1 Adobe Flash Player 的运行机制

1.1 Flash客户端技术分析

1.1.1 ActionScript语言

ActionScript 是一种面向 Adode Flash Player 的运行时编程语言，在应用程序和 Flash 内容中实现数据处理、交互性和其他一些功能。从 Flash 4.0 开始就已使用 ActionScript 语言，到了 Flash 5.0，ActionScript 不再是一种单纯的脚本语言，它已经发展为一种编程语言。在 Flash MX 2004 发布时，ActionScript 2.0 语言被推出，ActionScript 2.0 的最大特色是对数据类型进行严格定义，并且拥有了面向对象的编程模型 [4]。Flash Player 8.5 推出了 AVM2（ActionScript Virtual Machine2）虚拟机和 ActionScript 3.0 语言，AVM2 虚拟机用于执行 ActionScript 3.0 语言 [5]。

1.1.2 Flash技术框架

1）Flash 渲染模型研究

Flash 的渲染主要是在 Flash Player 中进行的。Flash Player 主要由 ActionScript 虚拟机（AVM）和图形渲染引擎（GR）两部分组成。其中，ActionScript 虚拟机用于对编译后的 ActionScript 字节码予以执行，图形渲染引擎用于对显示列表中的图形对象进行绘制。显示列表是一个树形结构的列表，它是由运行期间屏幕上渲染的图形对象所组成 [6]。只有显示列表中显示的对象才可以被渲染显示。图 1 描述了 Flash 运行时的显示列表。

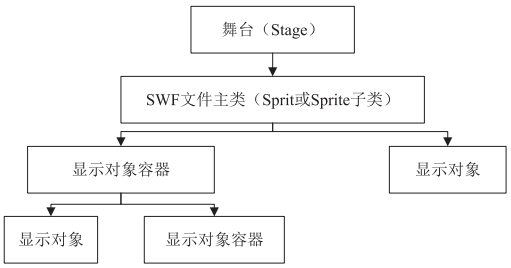

图1 Flash运行时的显示列表

图 1 中，显示列表的根是舞台对象（Stage），它是全部可视化对象的最外层容器。在 Flash Player 中，所有出现在应用程序屏幕上的图像元素都属于显示对象。

2）事件机制

Flash 的事件机制是以 DOM3 事件模型为基础的，通过对注册事件监听器进行创建，可以在对象的实例发出某个事件时，自动将该事件派发到所有注册过的监听器。对于 ActionScript3 来说，每个事件都是由事件对象来表示的。事件对象包含了一些操作方法和特定事件的信息 [7]。

1.2 Flash三维可视化分析

1.2.1 Flash三维图形显示原理

Flash 三维图形显示主要涉及三维坐标变换、纹理映

射和投影变换 3 个方面 [8]。

1）三维坐标变换

在 Flash 三维渲染显示时，点变换是三维形体变换的基础，对图形的一系列定点进行几何变换，然后将新的定点连接起来就可以产生新的图形。几何变换又可以分为缩放变换、旋转变换和平移变换。

2）纹理映射

纹理映射指在物体表面上映射纹理模式，可以通过矩形数组定义纹理模式。使用纹理映射能够制作出具有真实感的图形，而不需要对物体的表面细节做过多的考虑。

3）投影变换

使用投影变换可以在二维屏幕上投影三维物体，它的特点是物体离视点越远，成像便越小。投影模式可以进一步划分为正交投影（orthographic projection）和透视投影（perspective projection）两种。

1.2.2 Stage3D硬件加速

在 Flash Player 11 之前，Z 轴并不存在，Flash 只是对二维坐标进行显示。如果要模拟 3D 效果，需要先将三维坐标转换为二维坐标，然后再转换二维坐标为 Flash 显示时的坐标 [9]。Stage3D 可以通过 GPU 硬件加速对三维图形进行渲染，从而克服传统软件模式的束缚。

在 Stage3D 硬件加速时，我们只需要对几何形状进行定义，并为 GPU 传递数据，上载几何形状到 GPU 显存（GPU 的数据存储空间）。GPU 处理接收到的数据，它从分析顶点流开始，对三角形一个接一个地进行渲染，从而完成整个渲染 3D 内容的工作。这些步骤都是在显卡物理硬件内部完成的。GPU 通常对顶点集进行计算，并渲染由顶点集表述的三角形。这个任务是非常具体的，因此，3D 硬件渲染的过程是高效而迅速的 [10]。

1.2.3 Stage3D三维建模

在 Stage3D 中，通常一系列几何图形会组成需要被渲染的 3D 场景，而每个几何图形会被定义成一系列三角形，而每个三角形又会被定义成一系列顶点。

下面以构建正方形为例进行说明。首先通过一个向量（vector）定义所有顶点，其中每个顶点的数据包含顶点颜色和顶点坐标两部分；然后通过 Vertex Buffer 上传顶点数据到 GPU 内存；接下来利用 Index Buffer 对一个额外的顶点索引结构进行构建；最后传递这些内容到 GPU，从而完成对正方形几何模型的渲染。

2 SWF 文件格式分析

SWF 是一种支持点阵图形和矢量的动画文件格式。它采用流媒体技术，具有文件体积小、缩放不失真的特点，目前在动画制作和网页设计等领域得到了广泛应用。可以使用 Adobe Flash Player 打开 SWF，如果通过浏览器打开，必须对其安装 Adobe Flash Player 插件。SWF 文件由一个文件头和后面的一系列标签组成 [11]。标签又可分为定义型和控制型两类。其中，定义型标签将对象定义为角色，并把角色存储在字典中；控制型标签操作角色并控制流程。SWF 文件结构如图 2 所示。

图2 SWF文件结构

1）SWF 文件头

所有 SWF 文件都是从文件头开始的。在文件头的起始位置，标识 FWS 表示文件没有被压缩，标识 CWS 表示文件的第 8 个字节之后是通过 ZLIB 开放标准进行压缩的。文件头包括版本号、文件长度、画面尺寸、帧速和帧总数等信息 [12]。SWF 的文件头结构如表 1 所示。

表1 SWF文件头结构

区域	数据类型	含义
标识	UI8（8 位二进制无符号整数）	F 表示该文件是未压缩的 C 表示该文件是压缩的
标识	UI8	总是 W
标识	UI8	总是 S
版本	UI8	SWF 文件版本
文件长度	UI32	文件字节大小
画面尺寸	RECT	以 twips 为单位（单位帧的尺寸）
帧速	UI16	8.8 形式的定点小数
帧总数	UI16	总帧数

在 SWF 文件头中，版本号占一个字节，它是一个 8 位的数字而不是一个 ASCII 字符。文件长度对 SWF 文件的总长度进行记录，包括文件头。对于未压缩的 SWF 文件，文件长度即表示文件本身的长度；对于压缩的 SWF 文件，文件长度表示第 8 个字节之后的数据解压缩后的长度。画面尺寸对屏幕显示的宽度和高度进行定义，它通过 RECT 结构存储数据。在 RECT 结构中，Xmax 和 Ymax 分别定义宽度和高度，而 Xmin 和 Ymin 通常是 0。帧速对每秒的播放速率进行定义，帧总数则定义了 SWF 文件的总帧数。

2）标签

SWF 文件的标签有定义型和控制型两种。其中，定义型标签对 SWF 文件的内容（包括声音、位图、文本和 Shapes 等）进行定义。每一个标签会分配一个编号给定义的内容，这个编号被称为角色编号（character ID）且它是唯一的。定义型标签是不会产生渲染的。

控制型标签对字典中角色的实例进行创建和操作，并对文件的流程进行控制。SWF 文件的处理流程描述如下：Flash Player 在处理标签的过程中，当遇到 ShowFrame 标签时，就会将显示列表复制到屏幕，然后 Flash Player 继续对标签进行处理，直到遇见第二个 ShowFrame 标签。此时，第一帧记录了第一个 ShowFrame 标签之前全部控制型标签的执行结果，第二帧记录了从文件开始到第二个 ShowFrame 标签之间全部控制型标签的执行结果，依次类推 [13]。

3 Adobe Flash Player 漏洞分类

当前，Adobe Flash Player 软件得到了广泛应用，这也导致 Flash 漏洞种类繁多且具有较大的危害性。截至 2014 年 4 月，已经发布的 Adobe Flash Player 漏洞有 317 个，现对其进行分类 [14]。

1）Flash 文件格式漏洞

Flash 文件格式漏洞是一种常见的漏洞。攻击者利用这种漏洞可以对畸形的 Web 网页进行构建，使其包含恶意 Flash 文件，当用户对该页面进行访问时，则会触发 Flash 漏洞，从而使得攻击者能够以非法的方式访问用户进程，并在被入侵的系统中执行任意指令。Flash 文件格式漏洞通常是一种缓冲区溢出漏洞，根据溢出发生的位置，该种漏洞可以进一步划分为堆溢出漏洞和栈溢出漏洞两大类。

2）Flash 拒绝服务漏洞

Flash 拒绝服务攻击是指利用 Adobe Flash Player 的漏洞，采取伪装或欺骗的方式进行网络攻击，这种攻击方式会导致服务器因无法做出正确响应或资源耗尽而瘫痪，从而难以向用户提供正常服务。Flash 拒绝服务攻击具有简单有效、隐蔽性强的特点，在当前的网络攻击中得到了广泛应用。

3）Flash 跨站脚本攻击漏洞

攻击者在 Flash 文件中插入恶意 ActionScript 代码，通过 Web 页面加载该 Flash 文件，当用户对该页面进行浏览时，嵌入在 Flash 文件中的 ActionScript 代码会被执行，从而完成对攻击网站的访问。

4）Flash 欺骗攻击漏洞

在 Web 页面中，远程加载的 Flash 文件可能会对本地加载的 Flash 文件进行控制，从而欺骗用户，使其访问伪造的恶意网站，执行攻击代码。Flash 欺骗攻击通常会和其他技术结合使用。

对 CVE 漏洞库进行统计，可知将近一半的 Adobe Flash Player 漏洞是 Flash 文件格式漏洞。截至 2014 年 4 月，在已经发布的 317 个 Adobe Flash Player 漏洞中，Flash 文件格式漏洞 149 个，Flash 拒绝服务漏洞 76 个，Flash 跨站脚本攻击漏洞 38 个，Flash 欺骗攻击漏洞 10 个，其他漏洞 44 个。Adobe Flash Player 漏洞统计结果如图 3 所示。

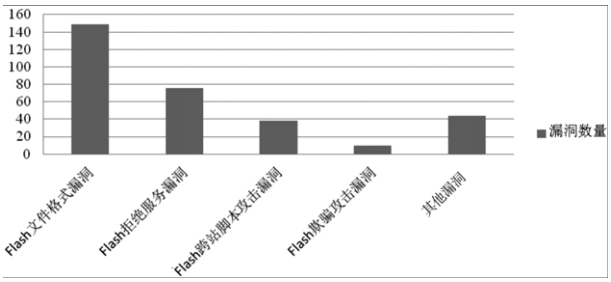

图3 Adobe Flash Player漏洞统计结果

4 Adobe Flash Player 漏洞分析方法

4.1 漏洞分析技术模型

漏洞分析技术通常是指分析已公开软件安全漏洞的机理，通过 POC 代码触发漏洞，对漏洞场景予以重现，并编写检测规则等。漏洞分析技术一般包括信息采集、分析调试以及漏洞利用分析 3 个阶段，通过对漏洞进行分析，最终可以建立检测规则，并为修复漏洞提供支持。

漏洞分析技术模型如图 4 所示。

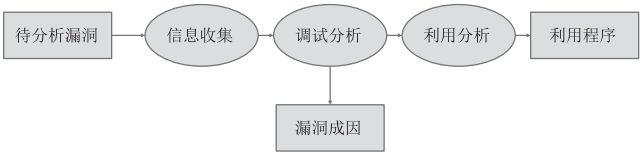

图4 漏洞分析技术模型

4.2 信息收集

Adobe Flash Player 漏洞分析技术中的一个重要环节就是信息收集，它主要是在调试分析之前，收集与漏洞相关的信息，包括漏洞公告信息、严重等级指数、异常信息和格式信息等 [15]。

1）漏洞公告信息

对 CVE、CVND 等国内外大型漏洞库和 Adobe 官方网站进行检索，分析它们发布的漏洞公告，获取漏洞类型、受影响软件的版本和名称以及漏洞细节等信息，同时利用网络资源对更多的信息进行收集，以为后续工作提供支持。

2）异常信息

Adobe Flash Player 在运行过程中出现异常时，监测此时程序的寄存器数据、堆栈数据以及进程和线程的上下文状态。

3）格式信息

主要涉及 SWF 文件的格式信息，需针对 Adobe Flash Player 的输入格式进行解析，从而在漏洞分析过程中更快地定位漏洞点。

4.3 调试分析

调试是 Adobe Flash Player 漏洞分析的核心环节，通过前期的信息采集，对触发漏洞的特定字段进行记录，找出该字段与漏洞点之间的对应关系。在调试过程中可能会出现异常，如内存访问失败等，此时，需要通过调试工具监测这些异常，动态跟踪程序在解析文件格式过程中的系统状态，以迭代和递归的方式对漏洞成因进行分析。在调试分析过程中，需关注以下几个问题。

1）动态调试工具的选择

在 Windows 平台下，常用的动态调试工具有 Immunity Debugger、WindDbg 和 OllyDbg。OllyDbg 插件丰富，具有较强的可扩展性，极大地提高了灵活性；WinDbg 不但能调试 ring3 级的应用程序，还能够对 ring0 级的内核程序进行调试。

2）数据流的跟踪与监测

程序在进行逻辑处理时，由输入到处理再到输出的执行路径被称为数据流。我们需要关注数据流在程序执行过程中的每一个节点中是怎样被修改和覆盖内存的，并记录寄存器值的变化。程序对内存和寄存器的不当处理常常是安全漏洞产生的直接原因。数据流的跟踪与监测可以通过捕获异常和设置断点两种方式予以实现。

（1）捕获异常。动态调试工具可以对异常事件进行捕捉，当运行程序出现异常时，程序的执行将被挂起。当 WinDbg 捕获到由缓冲区溢出而导致的异常时，会利用异常分发函数 KiUserExceptionDispatcher() 的参数信息，对异常出现时系统堆栈中的数据长度、参数位置和完整性等信息进行观察。

（2）设置断点。断点可以分为内存访问断点、硬件断点和软件断点等。通过对内存访问断点进行设置，可以在特定内存地址的数据出现变化时，将程序挂起，然后对此时的内存和寄存器信息进行查看和分析。上文中提到的 3 种动态调试工具均可以设置内存断点。

3）漏洞原理分析

漏洞点代码指的是程序中触发漏洞的代码片段，它通常是一个函数。对漏洞点代码予以确定是漏洞分析的重要目标。首先跟踪和监测数据流，采用迭代和递归的方法对其进行回溯分析，从而确定产生漏洞的代码片段的位置，再对漏洞点附近的代码进行仔细阅读，最终分析漏洞产生的原理。

4.4 利用分析

Adobe Flash Player 漏洞的利用分析以其触发条件和产生原理为基础。首先根据漏洞调试过程中的数据对漏洞类型予以确定，然后通过漏洞的触发环境对漏洞的利用条件进行确定，最后根据可利用程序对漏洞的危害等级予以评估 [16]。

5 典型 Adobe Flash Player 漏洞原理分析

利用本文第 4 部分描述的漏洞分析方法对 CVE-2011-0618、CVE-2011-0619、CVE-2011-0624、CVE-2011-0625、CVE-2011-0626、CVE-2011-0627、CVE-2011-2455、CVE-2011-2456、CVE-2011-2457 和 CVE-2011-2460 等 10 个典型的 Adobe Flash Player 漏洞进行分析，经过信息收集、数据流的跟踪与监测以及漏洞原理分析等过程，最终得到漏洞的产生机理。

1）CVE-2011-0618 漏洞原理分析

SWF 文件中可以嵌入用于动态计算的 ActionScript 3 代码。ActionScript 3 代码中定义了一些方法，在方法头部的定义中包含了预先定义的堆栈空间大小。

对于一个嵌入 ActionScript 3 代码的 SWF 文件来说，如果方法头部定义的堆栈空间是非法的，则会导致内存崩溃。在 POC 文件中，ActionScript 3 代码位于 DoABC 标签中，它可以生成一个新的 SWF 文件，这里称它为 embedded.swf 文件。这个文件包含了 ActionScript 3 代码中定义的方法，方法

头部定义了堆栈的范围：栈空间最大值和位置深度最深值。

ActionScript 3 定义的方法仅包含 pushshort 指令，它将一个短整型数值压入堆栈。在处理这个整型数值的过程中，为其赋予一个非法的内存地址，由此导致内存崩溃，与此同时，POC 文件崩溃。

2）CVE-2011-0619 漏洞原理分析

SWF 文件可以包含 DefineFont4 标签，用来内嵌文件的字体信息。这些字体保存在 OpenType Compact Font Format（CFF）中，CFF 的结构如表 2 所示。在 CFF 附近有一个表记录结构（Table Record），表记录结构包含偏移量，偏移量决定着附加数据的开始位置。

表2 CFF结构

类型	名称	描述
ULONG	tag	4字节的标识符
ULONG	ChenkSum	表的校验和
ULONG	Offset	距TrueType字体文件开始位置的偏移量
ULONG	length	表的长度

漏洞触发的原因是 Adobe Flash Player 没有对这个偏移量做有效性检查，攻击者可以任意改变这个值，使得内存可以被随意读取，从而导致 Adobe Flash Player 崩溃。

3）CVE-2011-0624 漏洞原理分析

ActionScript 字节码中存在一种相对跳转指令——jmp 指令。由于 Flash10a.ocx 没有对 jmp 指令后包含相对跳转长度的区域进行边界检查，致使 ActionScript 字节码中的某处 jmp 指令跳出了 ActionScript 字节码的内存区域，从而导致程序出错。

4）CVE-2011-0625 漏洞原理分析

当执行一个 ActionIf 指令时，Flash Player 没有做充分的边界检查，就不能确保跳转之后指令指针的值位于 ActionScript 代码数据中。尤其可能发生后向跳跃，使得指令指针移动到 ActionScript 代码前的地址。

5）CVE-2011-0626 漏洞原理分析

该漏洞是 Flash Player 中的一个空指针解引用漏洞。当一个未初始化的指针被解引用时就会出现空指针解引用漏洞。对于 DefineFont/DefineFont2/DefineFont3 标签来说，当它们拥有不能被充分读取的 OffsetTable 时，即被认为是畸形的。DefineFont/DefineFont2/DefineFont3 标签被解析后，DefineFontAlignZones 标签也将被解析，该标签引用 DefineFont/DefineFont2/DefineFont3 标签，就会触发漏洞。

6）CVE-2011-0627 漏洞原理分析

Flash 文件中可以包含 ActionScript3 的动态计算，ActionScript3 的特性之一是可以采用剩余参数，剩余参数指定了逗号分隔参数的方法。该漏洞就是当 ActionScript3 去处理特定情况下的剩余参数时发生的。JIT 编译器是否对代码进行优化取决于剩余参数是如何被访问的。当优化器遇到 ActionScript3 字节码指令 setlocal0、setlocal1、setlocal2 和 setlocal3 时，将检查存储剩余参数的本地寄存器是否被新值覆盖。如果被覆盖，则不对这个协议进行优化。但当之前所进行的类型检查被破坏后，JIT 编译器会认为剩余参数并没有被覆盖，从而造成误判。类型检查可以为访问长度属性提供保护，确保长度属性在一个 ArrayObject 对象中被调用，而不是在其他类型的对象中，但是优化器将移除对剩余参数的类型检查，此时攻击者就可以控制指令指针，从而触发漏洞。

下面我们对字节码层面上的剩余参数进行分析。ActionScript3 定义的方法如果在 _info::flags 字段上有一个标志（NEED_REST=0x04），表明这个方法支持剩余参数。此外，如果方法的 _info_flags 字段设置了 NEED_ARGUMENT 标志（即 NEED_ARGUMENT=0x01），并且所有的方法参数都是无类型的，那么该方法也可以支持剩余参数。以上两种情况都有可能触发漏洞。

7）CVE-2011-2455 漏洞原理分析

ActionScript 脚本语言可以对自己的命名空间、类以及类型进行创建。在 AVM2 虚拟机中，任何类型的变量都需要查找自身所属的命名空间，从而得到合适的对象去完成变量的初始化和分配操作。在 AVM2 虚拟机中，可以查找每一个变量类型的命名空间。对于每一个特定的类型，均存在与之对应的命名空间，然而，当命名空间数目多于一个时，则会出现问题。

在进行检查时，AVM2 虚拟机仅能保证命名空间查找的结果不为空（即类型能够找到），但不会对返回值进行检查。这就导致返回值 -1 可能被用做一个指针，当试图对地址 [-1+0x14] 进行读取时，出现错误。

8）CVE-2011-2456 漏洞原理分析

在高版本的 Adobe Flash Player 中，Stage3D 组件被引入，这就为 Flash Player 增加了一些 3D 特性，其中一个特性是通过 Stage3D 格式处理数据，它被称为 Adobe 结构

格式（ATF）。ATF 是一个泛型容器，它可以被 Stage3D 组件的许多结构类型所使用。ATF 中的数据可以通过 uploadCompressedTextureFromByteArray() 方法被加载入 Flash Player。uploadCompressedTextureFromByteArray() 方法有 3 个参数，分别是 data（ByteArray 类型）、byteArrayOffset（unit 类型）和 async（Boolean 类型，初始为 false）。当解码和加载 ATF 数据时，ATF 头并没有被用于验证 ATF 数据大小和 byteArrayOffset 参数，这就可能导致处理这些数据时，指针指向了 ATF 数据缓冲区之外的数据，使得程序崩溃。

9）CVE-2011-2457 漏洞原理分析

ActionScript2 是以 Action 标签的形式存在于 .swf 文件中的，这些标签有多种类型，其中一些标签允许自定义函数（包括函数中的指令），另外一些标签允许调用之前定义的函数。这些标签在运行时被 Flash Player 解析。ActionScript2 中是允许递归的，这就使得 ActionScript2 解析器能够递归地调用一个内部函数。在 POC 文件中，该内部函数是 DoCallFunction()，它是被 DoAction() 调用的。

实际当中为了避免 X86 处理器的堆栈溢出，ActionScript2 对递归做了一些限制，它约束了每个栈帧的最长脚本运行时间以及最大递归深度。然而，如果 .swf 文件快速执行了多个迭代，并且每次迭代时均在栈上分配一个大的数据，那么就有可能触发漏洞。此外，虽然 ActionScript2 对递归做了相应的限制，但是 POC 文件在超时发生和递归深度检查之前可能就已经将堆栈资源耗尽，而 DoActions() 函数并没有对当前可用的堆栈空间进行检测，从而导致漏洞发生。

10）CVE-2011-2460 漏洞原理分析

由于 Adobe Flash Player 在早期版本中没有使用 Stage3D 机制，为了对 3D 图形进行处理，需要使用栈上静态大小的缓冲区对三维坐标进行存储。在这种情况下，被裁剪的表面会有更多的点被处理，就会出现缓冲区溢出，导致存储在邻接缓冲区的一个指针被覆盖。为了从内存区域读取数据，被损坏的指针后来被用做一个基值，该基值可以添加静态偏移。当三维表面截取了视口的所有边缘时，这个损坏的指针被处理，由此产生一个 9 个顶点的多边形，它是由 8 个顶点的多边形变化而来的。

将对象转换成三维坐标，并实时处理用户输入的旋转操作等都是通过 ActionScript3 完成的。在这一过程中，ActionScript3 创建了多个阶段，并修改对象的 Z 属性，使其成为 3D 图像。当有鼠标事件发生时，ActionScript3 会将该事件转换成三维旋转，即将鼠标 Display 对象的 X/Y 属性转换成三维坐标的 X/Y 属性。

在 CVE-2011-2460 中，裁剪的函数被调用了 5 次，分别是 near Z，near X，far X，near Y 和 far Y。每一个调用都可能增加最终裁剪输出的多边形点数。假如最终多边形点数大于等于 9 个顶点，覆盖将会发生。

6 结束语

本文首先从 Flash 客户端技术和 Flash 三维可视化分析两个方面对 Adobe Flash Player 的运行机制进行简单介绍，描述了 SWF 文件的格式信息，然后对 Adobe Flash Player 的漏洞分类方法和漏洞分析技术进行详细介绍，并以 10 个典型的 Adobe Flash Player 漏洞作为实例，详细分析了漏洞的产生机理。（责编 马珂）

参考文献

[1] 贺拓 .Flash 应用程序漏洞挖掘与利用 [D]. 西安 : 西安电子科技大学，2010.

[2] 刘海燕，杨洪路，王崛 .C 源代码静态安全检查技术 [J]. 计算机工程，2004，（30）：28-30.

[3] 岳彩松 .MS Office 漏洞挖掘与利用技术研究 [D]. 上海 : 上海交通大学 .2008.

[4] 朱虹丁，雁林 . 缓冲区溢出检测模型研究 [J]. 计算机工程与应用 ,2006,(6):25.

[5] 尚明磊，黄皓 . 缓冲区溢出攻击的分析与实时检测 [J]. 计算机工程 ,2005,(6):111-116.

[6] 胡奇光 . 基于 Flash ActionScript 3.0 的动画设计的研究 [J]. 计算机与数字工程 ,2010，38（7）:147-150.

[7] 孙颖 .Flash ActionScript 3 殿堂之路 [M]. 北京 : 电子工业出版社，2007.

[8] 吕辉 .Flash/Flex ActionScript 3.0 交互式开发详解 [M]. 北京 : 电子工业出版社，2008.

[9] 刘菲，孟祥增 .Flash 动画的内容特征分析与图像信息提取研究 [J]. 现代教育技术 .2009，19（12）：91-94.

[10] 刘菲 .Flash 动画的场景结构与视觉特征研究 [D]. 济南 : 山东师范大学 .2010.

[11] 袁江 . 基于 CVE 知识库的危急漏洞挖掘与分析技术研究 [D]. 哈尔滨 : 哈尔滨理工大学，2008.

[12] 王清 .0day: 软件漏洞分析技术 [M]. 北京 ：电子工业出版社 .2008.

[13] 段刚 . 加密与解密 (第三版)[M]. 北京 ：电子工业出版社 ,2008.

[14] 王晓飞 . 软件安全漏洞发掘技术研究 [D]. 长沙 : 国防科学技术大学，2006.

[15] 李畅 . 基于 Flash 的二维游戏设计 [D]. 北京 : 北京林业大学，2010.

[16] 刘小珍 . 基于 AVM2 逃逸的漏洞挖掘技术研究与防范 [D]. 成都 : 四川师范大学，2012.