安全威胁每周警讯

2012/12/02～2012/12/08

本周威胁指数

***TrendMicro* 中国区网络安全监控中心**

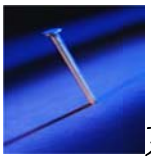
前十大病毒警讯

排名	病毒名称	威胁类型	风险等级	趋势	病毒行为描述
1	WORM_DOWNAD.AD	蠕虫	★★★★	→	该病毒会攻击未安装微软 IE 漏洞 MS08-067 的电脑，并且会在受感染电脑产生五万个恶意程序网址并试图在同一时间内随机连结其中 500 个恶意网站下载病毒
2	WORM_DOWNAD	蠕虫	★★★★	↑	该病毒会攻击未安装微软 IE 漏洞 MS08-067 的电脑，并且会在受感染电脑产生五万个恶意程序网址并试图在同一时间内随机连结其中 500 个恶意网站下载病毒
3	X97M_OLEMAL.A	宏病毒	★★	↓	宏病毒，它会将本身的下列副本放置到受影响的系统：%User Profile%\Application Data\Microsoft\Excel\XLSTART\k4.xls
4	TROJ_DOWNAD.INF	木马	★★★	→	DOWNAD 关联木马
5	TROJ_IFRAME.CP	木马	★★★	→	GIF、jpg 和 SWF 文件中被插入一个恶意的 iframe 标记时，趋势科技会将其判断为 TROJ_IFRAME.CP 病毒。当这些文件被执行时，会重定向到这些 URL，并下载恶意程序
6	X97M_LAROUX.BK	宏病毒	★★	→	宏病毒，主要通过用户浏览恶意网站感染，该病毒主要感染 Office Excel 文件，并且会在 Excel 中创建一个名为 StartUp 的宏脚本
7	WORM_ECODE.E-CN	蠕虫	★★★★	→	E 语言蠕虫,会在文件夹下建立同名 exe 文件
8	TROJ_FLDSCN.A-CN	木马	★★	↑	木马
9	X97M_LAROUX.CO	宏病毒	★★	↓	宏病毒，主要通过用户浏览恶意网站感染，该病毒主要感染 Office Excel 文件，并且会在 Excel 中创建一个名为 StartUp 的宏脚本
10	XF_HELPOPY.ARL	木马	★★	↑	木马，它会将本身的下列副本放置到受影响的系统：%Program Files%\Microsoft Office\OFFICE11\XLSTART\Book1

系统漏洞信息

MS12-073 ： Microsoft Internet Information Services (IIS) 中的漏洞可能允许信息泄露 (2733829)

Windows Vista
Windows Server 2008
Windows Server 2008 R2
Windows 7
描述：http://technet.microsoft.com/zh-cn/security/bulletin/MS12-073



系统安全技巧

Android 设备的权限机制设计，是为了让未事先宣告并取得权限的应用无法做出任何可能伤害设备的行为。但真的是如此吗？

Android 权限机制的运作方式

在深入探讨进一步细节前，让我们先来看看 Android 的权限机制如何运作。

Android 的应用对系统资源的访问是受到管制的。要访问一些敏感的应用程序编程接口，应用首先必须在 AndroidManiflest.XML 文件中宣告它所需要的权限。这些所谓敏感的 API 包括相机功能、GPS 定位信息、蓝牙与电话功能、短信/彩信功能，以及网络/数据访问功能。

在安装应用时，程序安装器会将这些程序宣告的权限显示给用户看，由用户决定是否接受并安装该应用。

若用户决定接受，那这些权限就永久授予给该程序，直到它卸载为止。运行时，系统不会再通知用户这些程序正在使用一些敏感的 API。反之，若用户决定不接受，那应用程序就无法安装。

假设有某个应用尝试使用这些受保护的功能，却未事先宣告，系统在执行该应用程序时就会出现安全性错误，并终止应用。

在这样的设计下，要避开权限机制似乎不太可能。但不幸的是，一些聪明的程序开发人员就是有办法开发出一些能避开这些权限机制的应用，方法就是滥用某些功能。

滥用默认浏览器并上传信息

在 Android 系统中，一个应用可以调用另一个应用的组件，然后通过一种名为“Intent”的抽象数据结构来描述所要执行的操作。每个 Intent 数据结构中都包含了要执行的动作，以及执行该动作需要的数据。

当一个应用发出 Intent 时，移动设备的操作系统会选择一个适合的应用程序来处理该数据。

例如，一个内含“Intent.ACTION_VIEW”这个查看动作的 Intent，再配上“Uri.parse(“http://www.google.com”)”这串数据，就代表该应用希望查看 Google 首页。当此数据结构传出来时，操作系统会决定该启动哪个浏览器。

在这样的设计下，意图不良的程序开发人员就能在自己的应用中通过 Intent 数据结构来启动浏览器，并将窃取到的数据上传至远程服务器。

由于是通过浏览器来访问这个网址，因此恶意应用自己根本不需要宣告 android.permission.INTERNET 这一访问互联网所必须的权限，因为浏览器已具备。

Logcat：功能远超乎您想象

另一种滥用系统权限机制的方式是通过 Logcat 这个系统内建工具。Logcat 是 Android 架构中用来记录系统事件的程序，可用来搜集、查看和过滤系统与应用的错误信息。系统同时也提供了一组 API，让应用开发人员将自己的除错信息写入其中。

要读取这些记录，应用只取得 android.permission.READ_LOG 读取日志文件的权限即可，而这项权限对一般用户来说貌似没什么危害。

程序开发人员可以将任何信息写入 Logcat，大多数时候写入的都是一些程序排错用的信息。尽管这些信息有时可能是一些敏感的信息（例如登录账号密码、信用卡数据），但粗心的程序员在应用发布之前，很可能会忘了将这些数据关闭。如此一来，一些心怀不轨的应用就能从日志文件中读取到这类信息，并将这些信息用于后续的攻击。

同样，操作系统也会将某些信息写入日志文件当中，因此也可能让歹徒有机可乘。

例如，在某些版本的 Android 系统中，当使用默认浏览器访问网站时，Logcat 就会记录所访问的网址。这样的动作看似无害，但恶意软件却能监控日志文件的内容，并从中提取各种网址，进而发掘用户喜好。通过这样的监控，坏人就能知道用户最常访问哪些网站，进而知道自己应该假冒哪些网站来进行网络钓鱼攻击。只要用户再度浏览同样的网站，应用还可能屏蔽该网站，并偷偷加载假冒的钓鱼网站。

另一个例子是 GPS 定位。某些应用在使用系统内的 GPS 定位服务时，会将当前的 GPS 定位数据写入日志文件。如此一来，恶意软件只要查询日志文件中的数据，就能追踪用户的最新位置。

滥用受保护的组件

Android 应用通常由四种组件组成：操作、服务、内容提供商，以及广播内容接收方。因此一个应用可能会通过多个进入点将其启动。要保护这些组件，防止它们遭到别的应用滥用，Android 建立了一套权限机制。

例如，通讯录信息是由系统所提供，应用需通过“内容提供商”查询其数据，而要查询这项数据，应用

必须先取得 android.permission.READ_CONTACTS 这一用于访问通讯录的权限，随后才能调用对应的功能。同样，应用必须先宣告 android.permission.INSTALL_PACKAGES 这一用于安装数据包的权限，才能自动安装别的应用。除了系统预先定义的权限外，应用也可以注册自己专属的权限。无论应用宣告了哪些权限，要调用别的程序，那么这个应用就必须先取得对应的权限才能执行调用。

但是目前仍旧有太多的应用程序在调用时并不要求任何权限。这些通常是设计给某个应用内部自己使用的组件。一些不够谨慎的开发人员或许认为，这些是私人用途的组件，没有其他应用（或歹徒）会知道它们的存在。

不幸的是，这却让恶意应用的开发人员有机可乘。这类程序设计者可能会刻意寻找所有隐藏或私人用途的组件，然后针对特定应用进行逆向工程，并通过 Intent 数据结构进行调用。

如此一来，若是有某个系统应用在设计上出现瑕疵，将造成很大的问题。系统预装的应用通常具备强大的功能，例如安装应用程序、读取敏感信息，甚至清除设备上储存的所有数据。此外，除非先将设备破解，否则用户也无法删除这些系统程序。

正如信息安全研究专家 Andre Moulu 所发现的，Samsung Galaxy S3 设备预装的很多应用程序都有严重的功能漏洞。根据他的研究，某个服务具备了从某个目录安装 APK 文件的能力，但却未贯彻权限机制，也不会检查调用方是否获得权限。

Moulu 还发现，还有另一个服务会将 APK 文件复制到 SD 卡内的某个目录，但也同样不会贯彻权限机制或检查调用方。通过这两项漏洞，恶意应用就能将自己的恶意 APK 文件放入对应目录，进而安装到系统内，不仅神不知鬼不觉，当而且也不需要取得任何系统权限。

来源：趋势科技博客