软件安全入门

第八课 恶意代码检测与分析

目录
CONTENTS

01 恶意代码分析基础

02 静态分析

03 动态分析

01 恶意代码分析基础

主要内容包括恶意代码分类和恶意代码分析的目的及流程。

传统上，按照恶意代码的功能可分为病毒、蠕虫、木马、后门、僵尸程序、下载器、Rootkit、恐吓软件、勒索软件等。

- ✓ 病毒：编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据，影响计算机使用，并能自我复制的一组计算机指令或者程序代码。
- ✓ 蠕虫：蠕虫也在计算机之间自我复制，但不依赖于寄生的宿主文件，可独自传播。
- ✓ 木马：指那些表面上有用的软件，实际目的却是危害计算机安全并破坏计算机的程序。
- ✓ 后门：指为攻击者提供不需要授权或采用攻击者预置的授权就可访问受害者系统的程序。

分析一个恶意代码，目的是为了分析恶意代码的攻击过程、功能作用，评估恶意代码造成的破坏和影响，提取恶意代码检测的特征，修补恶意代码利用的漏洞，防范类似安全事件的再次发生。典型的分析流程包括：

I. 静态属性分析。用成熟工具进行快速的静态属性分析，识别已经有分析结果的样本，避免重复分析。

II. 静态分析。对样本文件代码进行反汇编分析，识别其中的混淆代码，了解代码大致的结构的执行流程。

III. 动态分析。根据静态分析的结果为动态分析准备合适的分析环境，选取合适的分析方法和工具，执行动态分析，观察恶意代码的行为和功能。

IV. 根据静态分析和动态分析的结果，对恶意代码的攻击过程，功能特点、实现机理等关键点进行深入分析。

02

静态分析

主要内容包括杀毒软件扫描、文本类型确定、文件哈希计算、字符串信息提取、文件元数据提取、混淆代码识别、代码反汇编等。

2.1 静态分析方法(一)

杀毒软件扫描

杀毒软件主要是基于代码特征、数据特征等构成的病毒库进行匹配和检测。一般采用多个杀毒软件同时检测，如VirusTotal。

文本类型确定

可以通过文件后缀名或者利用工具，如file来确定可疑文件的文本类型。

文件哈希计算

利用样本文件的哈希值可以 识别相同的恶意代码，避免重复分析。常见的哈希算法包括循环冗余校验算法CRC32、消息摘要算法MD5、安全哈希算法SGA1等。

字符串信息提取

恶意代码中包含的文本字符串信息可能包含恶意代码类型、功能、用途等有用信息，可使用工具如Linux系统中的strings命令提取恶意代码中的字符串信息。

2.2 静态分析方法(二)

文件元数据提取

根据文件类型可提取相应的文件元数据信息，如PE文件可提取文件头信息从而获得文件头大小、入口点地址、是否加壳等信息。

混淆代码识别

一般先识别恶意代码采用的是哪种代码保护技术和工具，从而根据具体的代码保护算法进行相应处理。常用代码保护技术推断工具有PEiD、Sigbuster等。

代码反汇编

在获得恶意代码的真实代码后，可以通过反汇编分析查看恶意代码的执行逻辑和基本功能。常用工具有radare、IDA等。

03

动态分析

主要内容包括动态分析环境构建、动态行为分析、动态调试分析、反虚拟化分析对抗、反自动化分析对抗等。

在对恶意代码进行动态分析时，需确保恶意代码不对外部环境产生破坏和影响，因此通常是在受控的隔离环境里进行动态分析。

目前常用的构建受控的隔离环境的技术有三种：

沙箱程序

是一种按照用户制定的安全策略限制其他程序行为的执行环境。典型的沙箱程序包括Sandboxie、影子系统PowerShadow等。不能限制采用高级攻击技术的恶意代码，如rootkit、内核漏洞利用等。

虚拟化软件

虚拟化软件提供了跨系统、跨平台的虚拟化能力，同时具有良好的运行效率。常用的虚拟化软件包括VMWare和VirtualBox等。但无法应对需要构建跨系统架构的虚拟化环境或对代码运行过程进行干预的场景。

硬件模拟器

能够通过软件技术模拟出物理主机上并不真实存在的各种硬件设备，典型的硬件模拟器包括QEMU、Bochs等。但运行效率和易用性较差。

3.2 动态行为分析

观察可疑代码在实际运行时表现出的各种行为，是确定代码具有何种功能，评估代码是否具有恶意性的重要手段。用于动态行为分析的工具有很多：

本地行为分析工具

一款常用的Windows系统分析工具是Sysinternal。其中的Process Monitor用于采集和显示Windows系统文件、注册表、进程等相关事件。Process Explorer是一个高级进程管理工具，可以提供一个进程的详细信息，包括图标、命令行参数、代码镜像路径等。

网络行为分析工具

一款常用的网络协议分析工具是Wireshark，可识别和解析数百种协议规范，支持实时流量捕获与离线流量分析。

在线行为分析工具

常用的在线恶意代码动态分析服务如Anubis、Joe Sandbox等，将可疑代码上传到这些网站，利用网站后台的动态分析系统对样本文件进行自动化动态行为分析。其中Anubis平台底层采用QEMU硬件模拟器，支持Windows平台可执行文件和Android平台APK文件的安全性分析。

动态调试是软件分析的重要手段，在恶意代码分析中也被大量使用。根据调试器的运行权限不同，调试器可以分为4种类型：Ring3级调试器、Ring0级调试器、模拟调试器和硬件调试器。动态调试工具有很多种，Windows系统上常用的有WinDbg、IDA Pro、OllyDbg等，Linux上常用的有gdb、IDB等。

动态调试分析的优缺点：

优点：可以获得程序动态运行时的真实行为信息，相比于静态分析更加具体和准确。

缺点：执行路径单一，无法全面分析程序的各种运行状态，造程序行为。

针对动态调试分析的缺点，学术界提出了三种思路：

第一种思路：把待分析程序放在不同的环境中运行，观察并比较在不同环境下程序行为表现的差异。

第二种思路：进行动态多路径分析。在程序执行到某一个分支点时保存系统状态，在执行完某个分支路径后将程序恢复到上一个保存的状态，并改变分支条件，重新选择未遍历过的分支路径。

第三种思路：强制程序执行特定的路径，观察该路径上是否出现恶意行为。如利用污点传播和符号执行等方法理解程序如何根据输入数据选择执行路径的特点，通过修改相应的数据使程序执行到不同的路径。

恶意代码为了避免自身被发现和分析，通常会采用一些反调试、反分析技术来对抗分析过程。

对抗直接的跟踪调试分析：通过检测操作系统相关接口如IsDebuggerPresent、OutputDebugString等获取自身是否被调试的信息，或通过检测父进程的名字、枚举当前系统中运行的其他进程以察觉自身是否被调试。

对抗虚拟化分析：通过分析执行环境与真实系统之间的差异，恶意代码可以察觉自身是否处于虚拟系统之中。如对于常见的VMWare虚拟机环境，可通过执行特权指令、利用中断描述符表基址检、存储任务寄存器STR内容等来检测虚拟机。

为了避免恶意代码检测到虚拟机环境，恶意代码分析人员需要对用于恶意代码分析的虚拟机进行修改和配置或在恶意代码放入虚拟机进行分析之前，对恶意代码是否包含虚拟机检测功能进行扫描。

3.5 反自动化分析对抗

为了提高分析效率和响应速度，各种恶意代码自动化分析技术被提出来。恶意代码编写者也在不断尝试采用各种技术来对抗自动化分析系统。反自动化分析对抗技术可分为如下几类：

1.**用户行为相关的对抗技术。**如通过检测运行环境中是否存在鼠标移动、键盘按键等消息事件，监视运行环境中是否存在光驱插入等用户操作，识别无人值守的自动化分析系统。

2.**分析系统环境相关的对抗技术。**通过检测运行环境的系统序列号、用户名、内存大小、硬盘大小等。

3.**分析系统软件相关的对抗技术。**通过查看分析系统是否安装常见应用软件、是否运行了特定的信息采集工具等。

4.**时间相关的对抗技术。**通过调用sleep系统函数、循环执行特定的代码段等拖延时间的操作，迫使自动化分析过程超时终止

5.**分析系统自身相关的对抗技术。**通过执行循环的加解密操作，重复调用大量无意义的系统调用等，迫使分析系统记录大量的分析过程数据，造成分析过载。

课程小结

- 1.恶意代码分析基础
- 主要内容包括：恶意代码分类和恶意代码分析的目的及流程。
- 2.静态分析
- 主要内容包括：杀毒软件扫描、文本类型确定、文件哈希计算、字符串信息提取、文件元数据提取、混淆代码识别、代码反汇编等。
- 3.动态分析
- 主要内容包括：动态分析环境构建、动态行为分析、动态调试分析、反虚拟化分析对抗、反自动化分析对抗等。

谢谢观赏