XCon®2010

AVM2虚拟机逃逸技术研究

刘小珍/dgguai27
dgguai27@hotmail.com
四川师范大学计算机科学学院
四川师范大学网络与通信技术研究所

议　程

- 问题的提出
- AVM2安全要点与潜在缺陷
- 典型逃逸方式与案例解析
- 防范策略
- 下一步工作

问题的提出

- HLL虚拟机逃逸
  - HLL虚拟机介绍

    .NET CLR、AVM2、JVM...
  - 逃逸的引出及研究范畴

    两个要素：虚拟机运行与安全漏洞
  - 逃逸技术研究价值

逃逸技术研究价值

通过虚拟机运行特性的辅助，研究人员可能会在相关漏洞的产生、分析利用和修补过程中获取传统研究方法所无法比拟的突破性思路或解决方案。

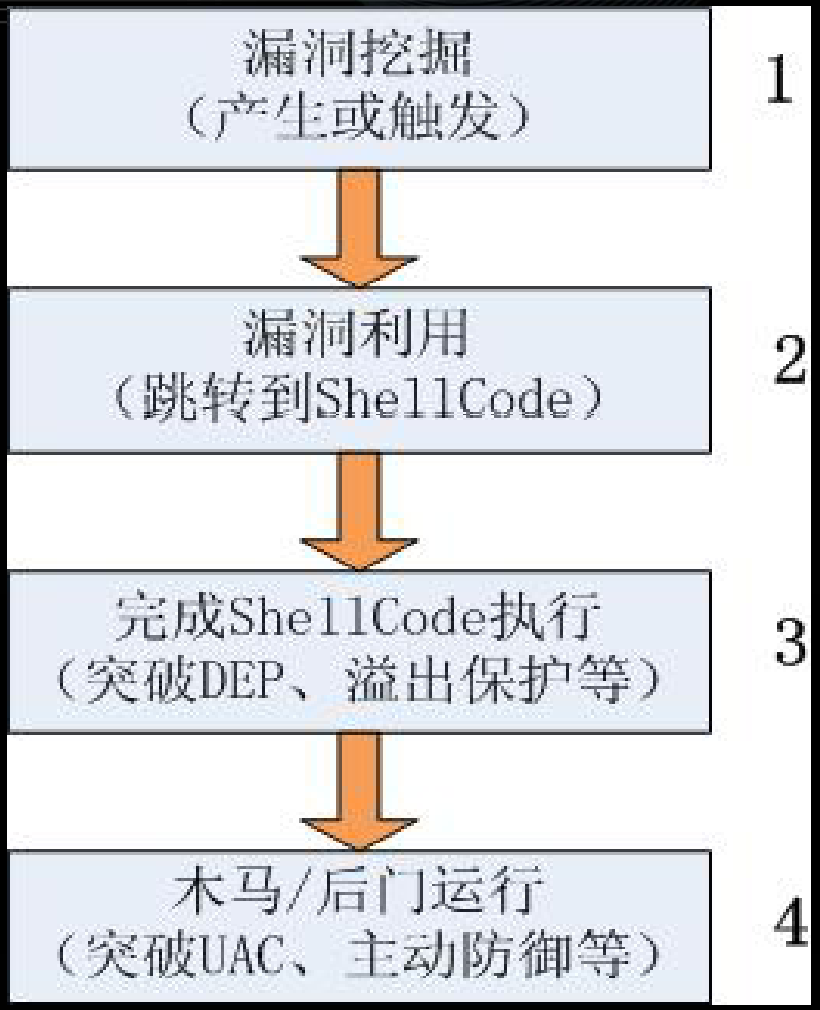
问题的提出

- 典型代表:AVM2逃逸
  - 相关知识回顾

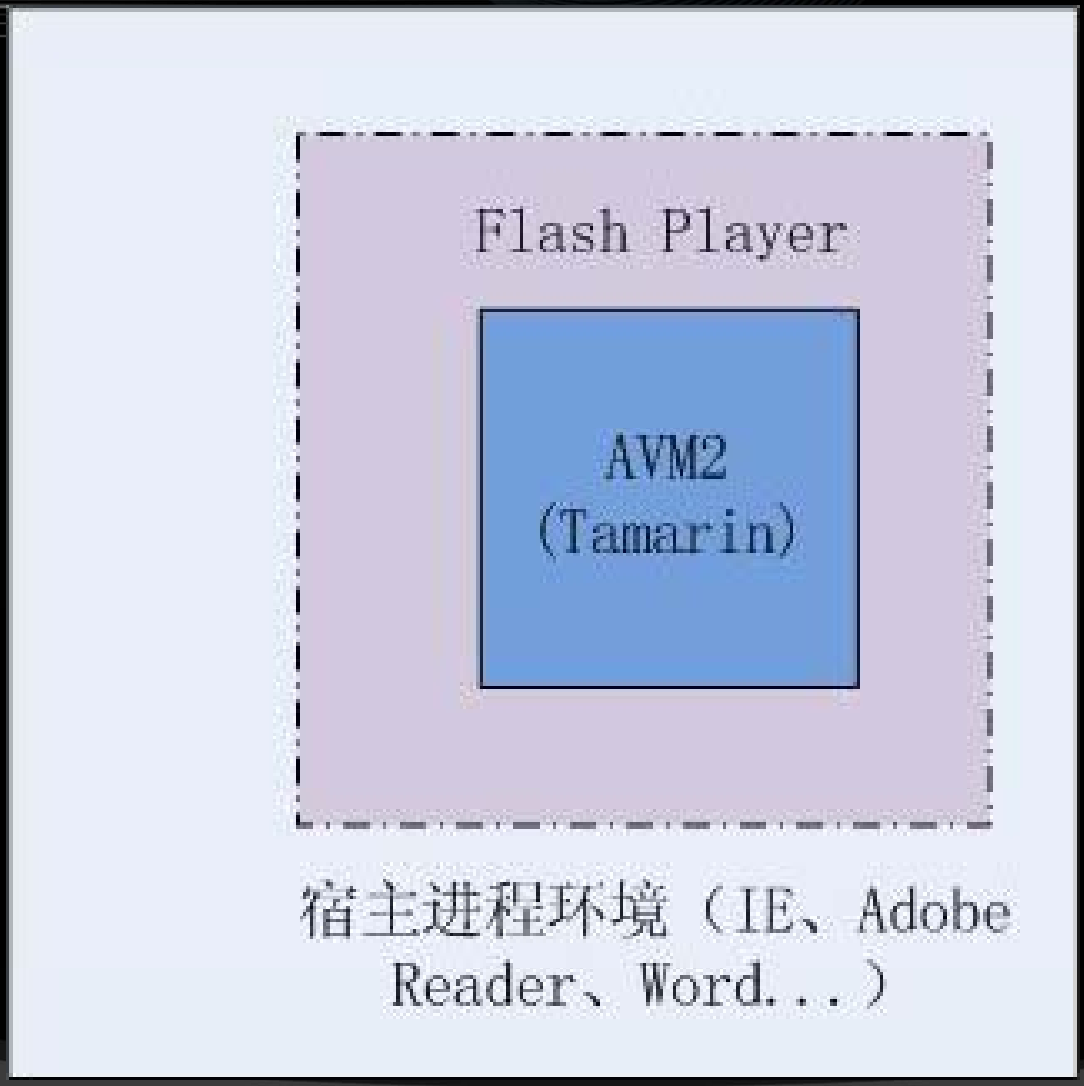
    漏洞攻击四层模型（FlashSky）

    AVM2运行环境—“寄生”，只考虑独立Flash作为控件
  - 逃逸模式

漏洞攻击四层模型


漏洞挖掘
（产生或触发）
1
漏洞利用
（跳转到ShellCode）
2
完成ShellCode执行
（突破DEP、溢出保护等）
3
木马/后门运行
（突破UAC、主动防御等）
4

AVM2的“寄生”方式



AVM2逃逸模式

结合AVM2运行过程或结果最终跳转到宿主环境中的ShellCode执行，或者在宿主环境中成功执行完了ShellCode（二者区别在于是否需要突破ShellCode执行时硬件DEP和溢出保护等障碍）。

内因：AVM2安全要点以及潜在缺陷

外因:安全漏洞

议　程

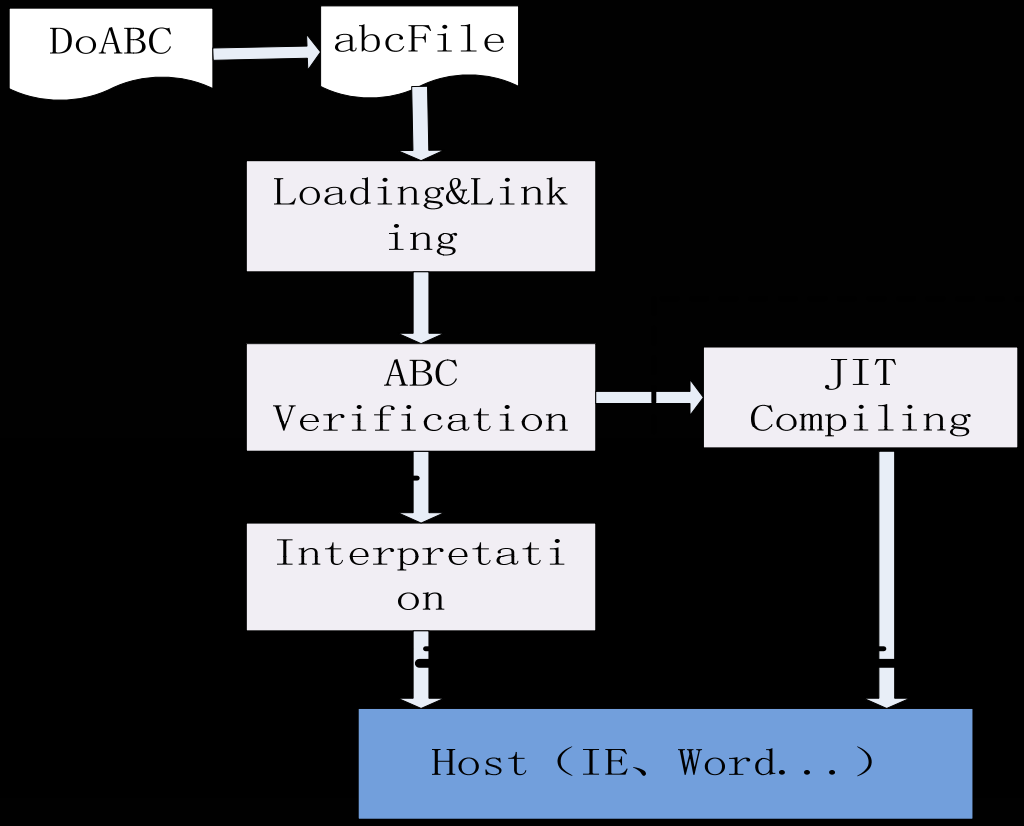
- 问题的提出
- AVM2安全要点与潜在缺陷
- 典型逃逸方式与案例解析
- 防范策略
- 下一步工作

AVM2安全要点

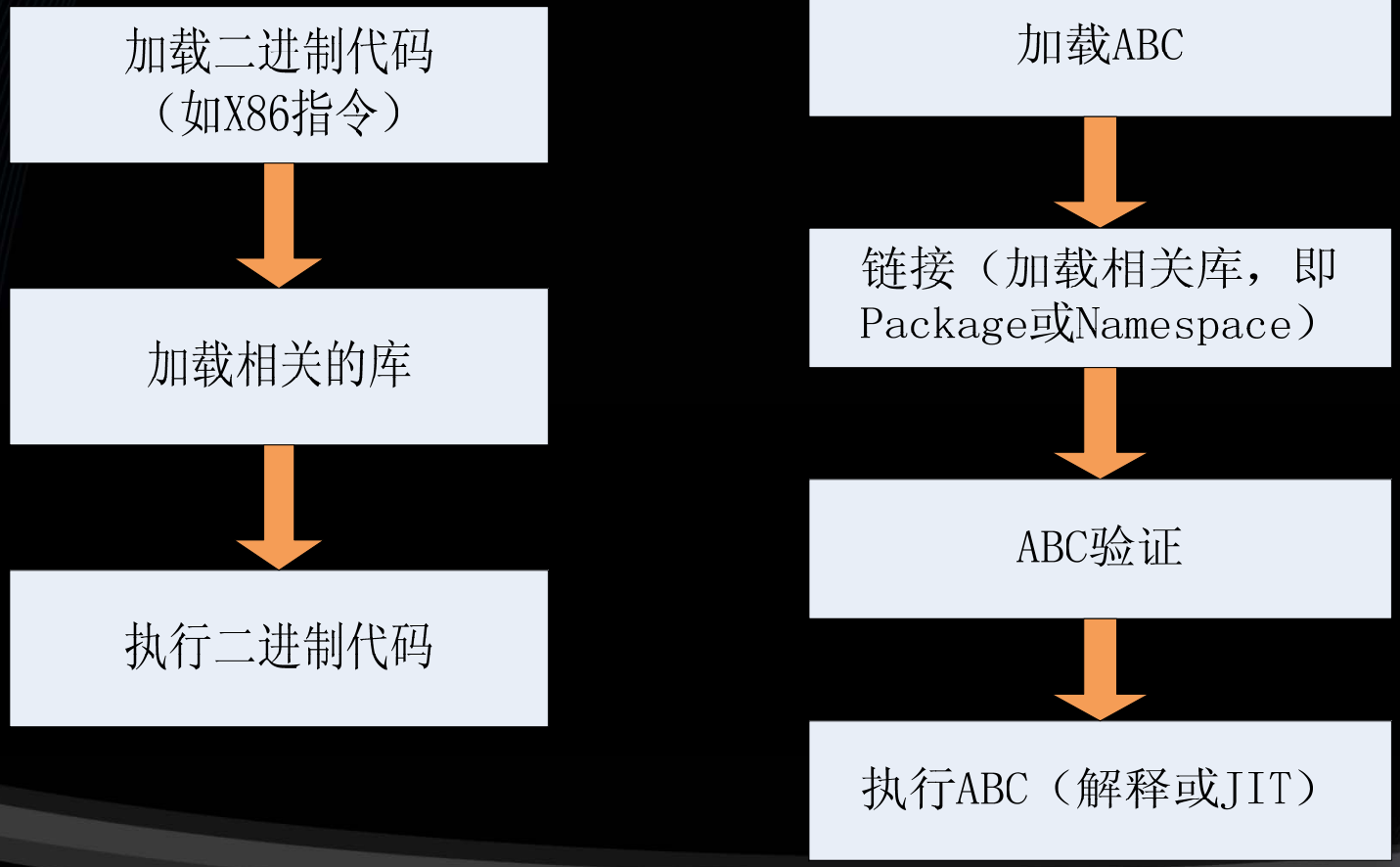
- ABC验证
- AVM2内存管理
- “寄生”模式

ABC验证

ABC执行流程:



ABC验证


加载二进制代码
（如X86指令）
加载相关的库
执行二进制代码
加载ABC
链接（加载相关库，即
Package或Namespace）
ABC验证
执行ABC（解释或JIT）

ABC验证

为字节码执行提供一个安全沙箱（Security Sandbox），使得不能在沙箱之外访问内存或执行代码；包括指令有效性检查、参数（或操作数）类型与值的有效性检查等，特别是用静态数据流分析对虚拟寄存器访问范围和转移指令目的地址的判断。

AVM2内存管理

本地数据区：

虚拟寄存器（Virtual Registers）、参数栈（Oprand Stack）和作用域栈（Scope Stack）

三者按地址从低到高依次从AVM2的运行时堆栈中分配 。

内部数据表示：Atom

“寄生”模式

依托于多种宿主进程环境

IE内核浏览器/Office: Flash9x.ocx、Flash10x.ocx

非IE内核浏览器：NPSWF32.dll

Adobe Reader/Acrobat: Authplay.dll

潜在缺陷

- ABC验证缺陷
- 内存管理缺陷
- “寄生”模式风险

ABC验证缺陷

- 验证依据的稳定性问题

  Flash9f及以下版本中Mask表存储在程序的可写数据段, 使得运行时可以动态改写

- 动态隐患的探测问题

  动态隐患，指的是AVM2执行字节码过程中出现的各种严重的内存破坏漏洞。

  uninit 、 use after free…

Flash9e.ocx使用的Mask表:

```
.data:302B3830  FF 00 00 00 01 01 01 00  01 00 FF FF 01 01 01 01
.data:302B3840  01 01 01 01 01 01 01 01  01 01 01 02 00 00 00 00
.data:302B3850  00 00 FF 00 01 01 00 00  00 00 00 00 01 01 01 01
.data:302B3860  00 01 02 FF FF FF FF FF  FF FF FF FF FF FF FF FF
.data:302B3870  01 01 01 02 02 02 02 00  00 01 02 FF 02 FF 02 02
.data:302B3880  FF FF FF FF FF 01 01 00  01 01 01 FF FF 01 01 01
.data:302B3890  01 01 01 01 00 01 01 FF  01 FF 01 FF 01 01 01 01
.data:302B38A0  00 00 00 00 00 00 00 00  00 FF FF FF FF FF FF FF
.data:302B38B0  01 00 00 00 00 00 01 00  00 00 FF FF FF FF FF FF
.data:302B38C0  00 00 01 00 01 00 00 00  FF FF FF FF FF FF FF FF
.data:302B38D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
.data:302B38E0  00 00 01 00 00 FF FF FF  FF FF FF FF FF FF FF FF
.data:302B38F0  00 00 01 01 00 00 00 00  FF FF FF FF FF FF FF FF
.data:302B3900  00 00 00 00 00 00 00 00  FF FF FF FF FF FF FF FF
.data:302B3910  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF 02 04
.data:302B3920  01 01 01 00 FF FF FF FF  FF FF FF FF FF FF FF FF
```

内存管理缺陷

本地数据区的分配方式，可能导致宿主环境中的某些代码指针（如函数返回地址）被覆盖。

“寄生”模式风险

AVM2依托于IE等多种宿主环境的
运行方式可能带来额外的逃逸威胁。

议　程

- 问题的提出
- AVM2安全要点与潜在缺陷
- 典型逃逸方式与案例解析
- 防范策略
- 下一步工作

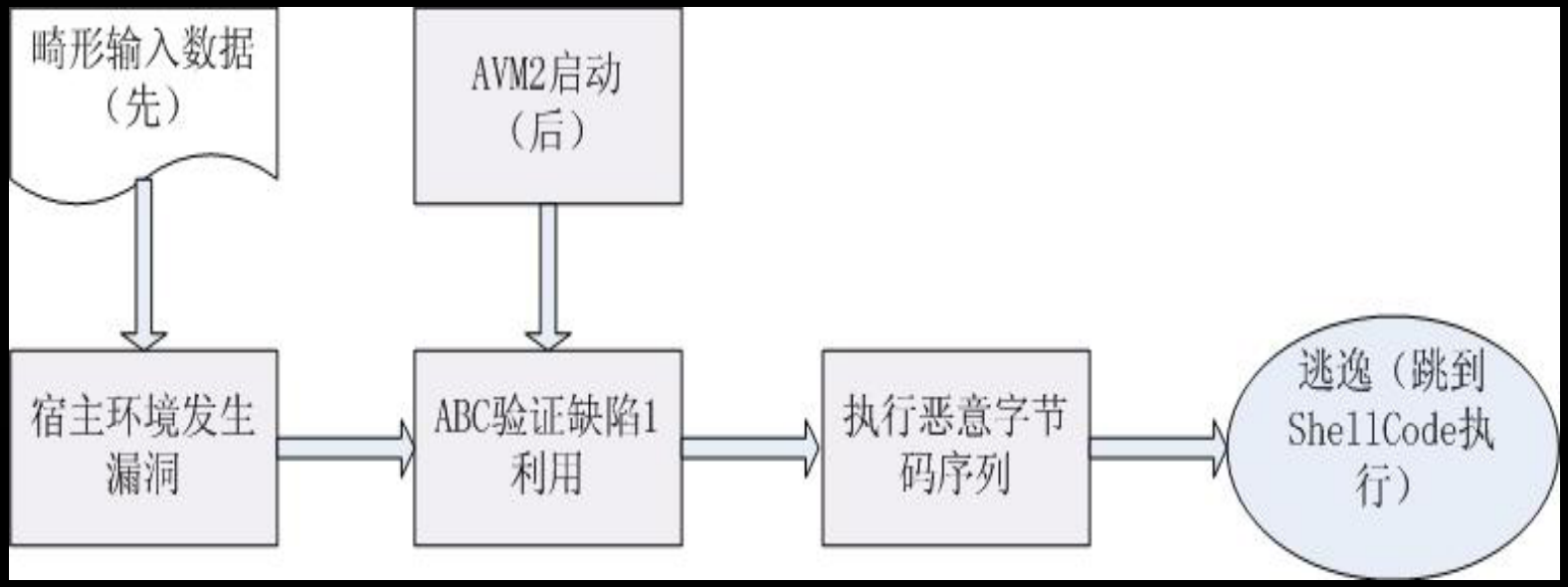
典型逃逸方式

- ABC验证缺陷利用I型
- ABC验证缺陷利用II型
- “寄生”模式风险利用
- 安全缺陷综合利用

ABC验证缺陷利用I型

宿主环境产生漏洞，结合AVM2运行过程中ABC验证缺陷1和内存管理缺陷实现逃逸，结果上只考虑跳转到ShellCode执行即可。

ABC验证缺陷利用I型



ABC验证缺陷利用I型

案例解析：CVE-2007-0071

- Dereference Null

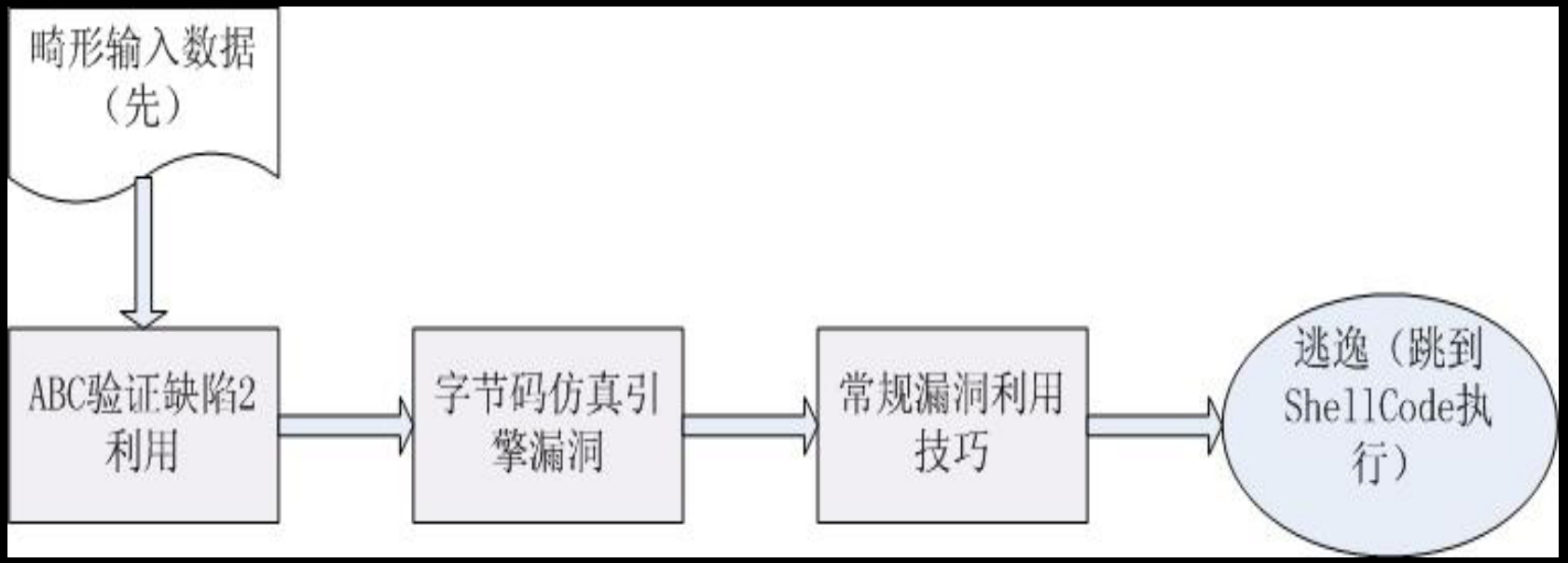
  常规方法难以利用

- 结合Mark Dowd方法

ABC验证缺陷利用II型

AVM2运行过程中由于ABC验证缺陷2而产生漏洞，结合常规漏洞利用技巧实现逃逸，结果上也只考虑跳转ShellCode执行即可。

ABC验证缺陷利用II型



ABC验证缺陷利用II型

案例解析：CVE-2009-1866

- 字节码解释引擎漏洞

25 81 10: push short

25 81 10: push short

C7: multiply_i

30: pushscope

5D 01: findpropstrict

- 结合常规漏洞利用技巧

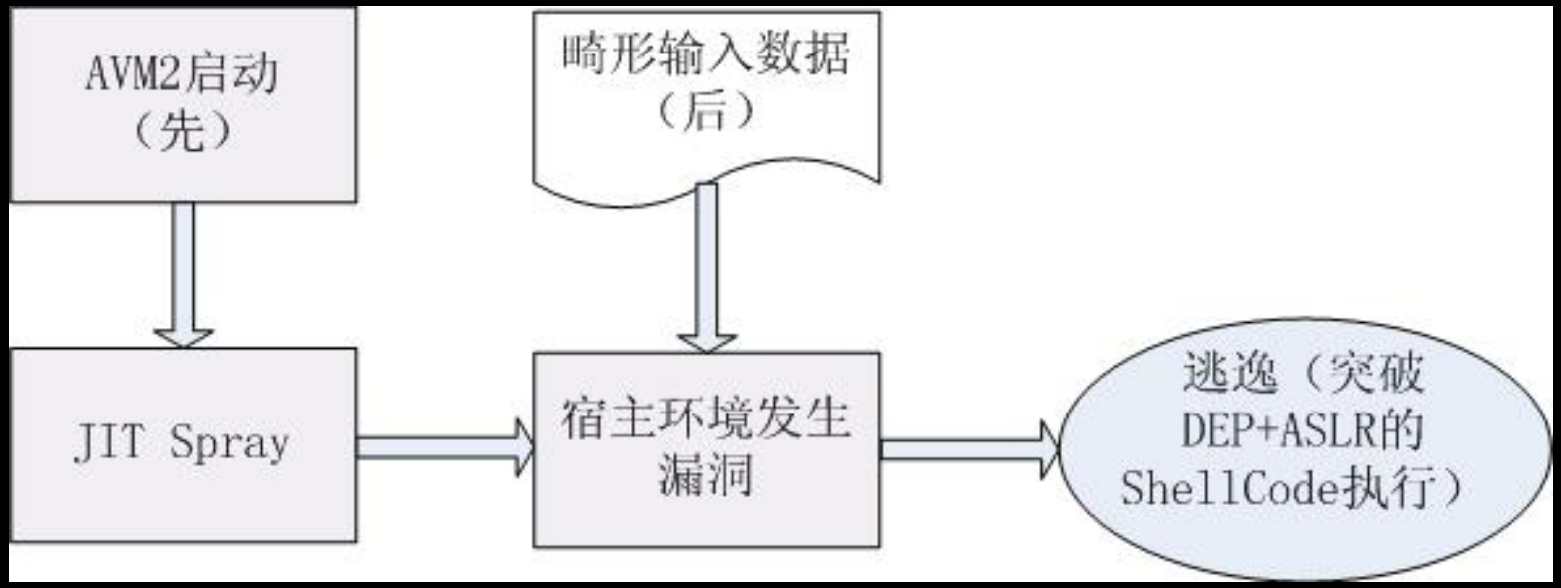
JS HeapSpray、AS HeapSpray

demo1

“寄生”模式风险利用

宿主环境产生常规漏洞，结合AVM2的特定运行结果（如JIT之后留在内存中的X86代码，其实这并非AVM2自身问题）实现逃逸突破，结果上需考虑ShellCode突破硬件DEP等障碍。

“寄生”模式风险利用


AVM2启动
（先）
畸形输入数据
（后）
JIT Spray
宿主环境发生
漏洞
逃逸（突破
DEP+ASLR的
ShellCode执行）

“寄生”模式风险利用

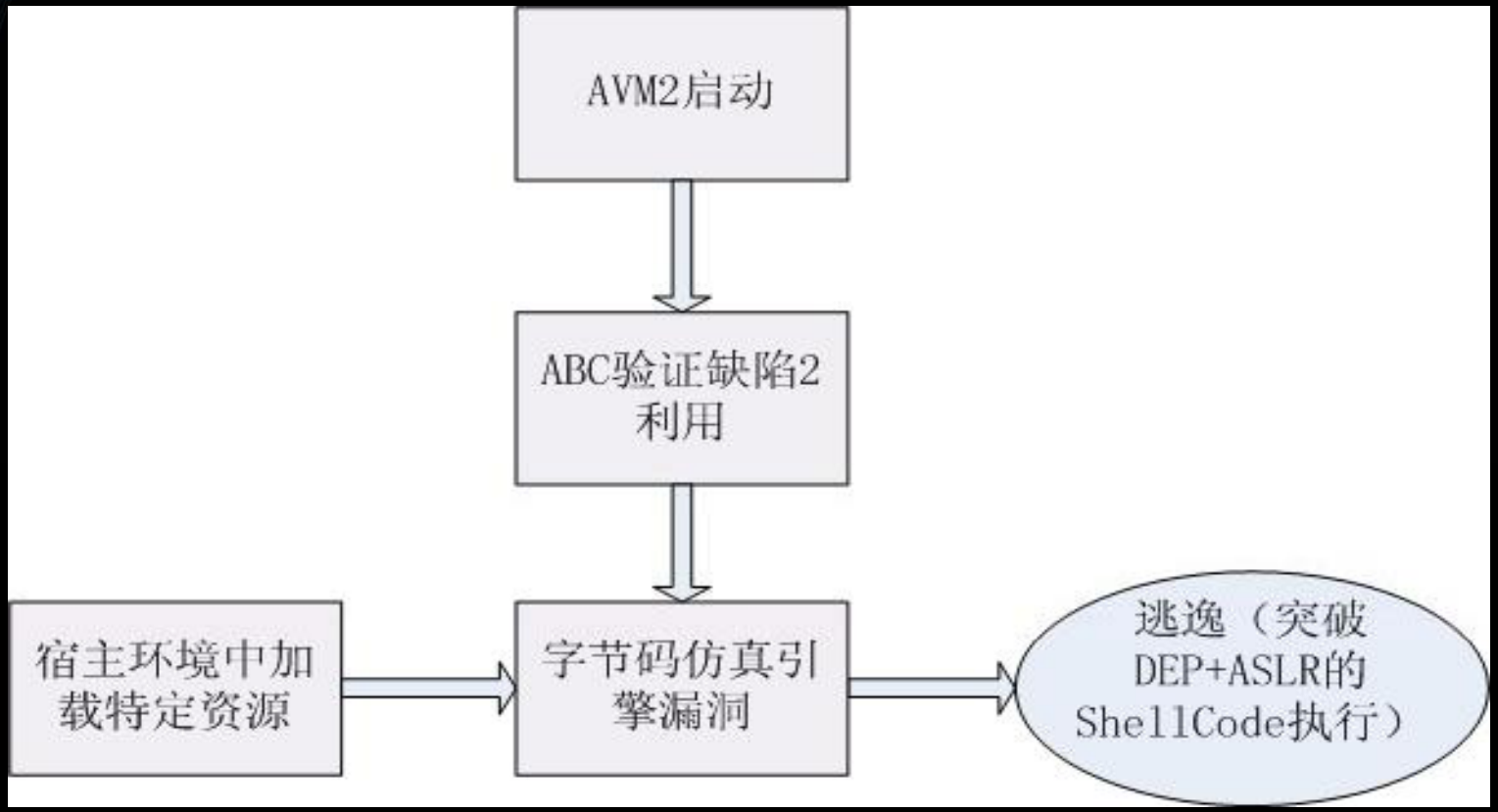
- AVM2对解释执行或JIT的选择
- IE8、许多ActiveX中的漏洞都可能通过本逃逸方法进行利用

Demo2（略）

安全缺陷综合利用

AVM2运行过程中由于ABC验证缺陷2而产生漏洞，同时需要结合宿主环境中特定资源实现逃逸，结果上要考虑ShellCode突破硬件DEP等障碍。

安全缺陷综合利用


AVM2启动
ABC验证缺陷2利用
宿主环境中加载特定资源
字节码仿真引擎漏洞
逃逸（突破DEP+ASLR的ShellCode执行）

安全缺陷综合利用

案例解析：CVE-2010-1297

- 字节码JIT编译过程漏洞

  newfunction处理问题， 未对40参数进行检测

  4e -> 40

- 结合BIB.dll+Ret-to-libc

demo3

议　程

- 问题的提出
- AVM2安全要点与潜在缺陷
- 典型逃逸方式与案例解析
- 防范策略
- 下一步工作

防范策略

- 更新系统中的Flash Player
- 主动开启系统的硬件DEP
- 在系统中安装Microsoft EMET系列工具
- 更新浏览器至IE8
- 更新系统到windows 7
- 采用Sandboxie等沙箱浏览器
- 安装主动防御软件

议　程

- 问题的提出
- AVM2安全要点与潜在缺陷
- 典型逃逸方式与案例解析
- 防范策略
- 下一步工作

下一步工作

- 继续深入分析Tamarin源码
- HLL虚拟机逃逸与Win7、Mac OS结合
- Flash漏洞挖掘技术的进一步研究

XCon®2010

谢谢大家

不足之处敬请指正