X'CON 2003

病毒和网络攻击中的多态、变形技术原理分析及对策

作者：Hume_

日期：2003.12

目录

X'CON 2003

- 怀疑者的目光—什么是多态和变形技术
  - 历史回顾
    - 游戏的开始—病毒技术简史
    - 游戏的结束？—特征码提取和反病毒技术的发展
    - 游戏仍在继续
  - 什么是多态（**Polymorphism**）
  - 什么是变形（**Metamorphism**）
- 恶作剧者的游戏？—多态和变形技术原理分析
  - 病毒多态和变形技术原理分析
    - 多态引擎
    - 变形引擎
  - 多态和变形技术在网络攻击中的应用

Copyright © Hume 2003

目录（续）

X'CON 2003

- 觉醒的人们—如何对抗多态和变形技术
  - 反病毒技术的演化
    - 防毒卡技术
    - 特征码提取和检测
    - 防患于未然——防火墙技术
    - 虚拟机技术和启发扫描
  - 入侵检测技术
- 未来的世界——人工智能？
- 结束

Copyright © Hume 2003

一、什么是多态和变形技术

1.1 历史回顾

- 游戏的开始——病毒技术简史
  - 五十年代末六十年代初。AT&T的贝尔实验室三个年轻的程序员制作了一个游戏：“磁芯大战” (core war)。
  - 七十年代上半叶，“爬行者”病毒通过网络（用猫链接的一对一的网络）进行传播。

Copyright © Hume 2003

游戏的开始—病毒技术简史

X'CON 2003

- “磁芯大战”之后，六十年代晚期到七十年代早期，在一种大型电脑—**Univax 1108**系统上，首次出现了和现代病毒本质上是一样的东西，一个叫做“流浪的野兽”（**Pervading Animal**）的程序可以将自己附着到其它程序的最后。
- 七十年代上半叶，在**Tenex(**一种叫做泰尼克斯**)**操作系统上出现了一种名为“爬行者”病毒，这个病毒可以通过网络（用猫链接的一对一的网络）进行传播 。
- 八十年代早期随着**BBS**的普及，以盗取账号和密码为目的的特洛伊木马、引导区病毒以及感染软盘
- **1988**年“莫里斯的蠕虫”（**Morris ‘s Worm**），第一个通过因特网传播的病毒出现。

Copyright © Hume 2003

Linux并非乐土

- **1990**年，第一个多态病毒“变色龙”**(Chameleon)**出现（又叫做“**V2P1**”、“**V2P2**”和“**V2P6**”）出现。
- **1995**年秋天“概念”（**Concept**）病毒开始在世界范围内流行，这一病毒的出现宣告了一种新形态的病毒的出现——宏病毒。
- **1997**年**2**月：第一个**Linux**环境下的病毒“上天的赐福”（**Bliss**）出现，**Linux**在此之前还是一个没有被病毒感染过的乐土。
- **1997**年**12**月：一种新的病毒形态，“**mIRC**蠕虫”出现。
- **1998**年**8**月：第一个感染**"**爪哇**"**（**Java**）可执行文件的病毒**"**陌生的酿造**"**（**Strange Brew**）问世。
- **1998**年**11**月：一种新的使用**VB**脚本语言编写的病毒“兔子”（**Rabbit**）。

Copyright © Hume 2003

- 九十年代到现在，病毒技术也日益完善，吸取了其它方面技术的成果，加密、变形等技术被病毒制造者运用的炉火纯青，跨平台感染的病毒也开始出现，大量的蠕虫和病毒使用高级语言病毒编写，宏病毒等脚本病毒大量涌现。利用漏洞的蠕虫病毒随着网络的发展也愈发猖獗，给全世界计算机用户造成了巨大的损失。蠕虫、病毒、木马之间的分界线也愈益模糊，一个病毒通常具有多种感染传播手段并具有多种意图。
- 分布式的、应用复杂人工智能的具有自我学习能力的智能型蠕虫或病毒亦可能会出现。

Copyright © Hume 2003

游戏的结束？—特征码提取及反病毒技术的发展

- 七十年代上半叶随着“爬行者”病毒的出现，一种叫做“清除者”的程序也被开发出来专门对付“爬行者”，这可能就是病毒和反病毒的第一次战争。
- **1989**年，俄罗斯程序员开始开发著名的反病毒软件**AVP**。
- 此后**KILL**、“赛门铁克”的诺顿、**McAfee**、瑞星、**KV**等杀毒软件相继出现。
- 在早期，反病毒技术就是特征码提取和查找的代名词，不过这给传统的病毒也带来了致命的打击。
- **1997**年出现了病毒防火墙技术。
- 反病毒厂商相应发展了虚拟机、启发式查毒等技术对抗未知病毒，也取得了很好的效果。
- 随着蠕虫类病毒的猖獗，反病毒技术和入侵检测等技术有融合的趋势。

Copyright © Hume 2003

游戏仍在继续

X'CON 2003

- 由聪明的程序员发起的这场游戏会结束吗？计算机的存在、程序的本质决定了这场攻防的战争永远也不会停止。
- 我们能做什么？
  - 了解这些技术，开发高技术含量的防毒软件、扩大市场并降低软件价格。

Copyright © Hume 2003

1.2 什么是多态（Polymorphism）

X'CON 2003

- 多态并不神秘。病毒多态就是使病毒能够改变自身的存储形式的技术，使传统依靠特征值检测的技术失效。

Copyright © Hume 2003

1.3 什么是变形(Metamorphism)

- 变形则在多态的基础上更进一步。对整个病毒体都进行处理，使不同病毒实例的代码完全不同，不但没有固定的特征码，而且也无需还原成没有任何变化的病毒体，如果说对付多态还可以通过虚拟机等待病毒体被还原之后检测特征值，那么完全的多态则使得这种技术完全失效。

Copyright © Hume 2003

二、恶作剧者的游戏？
—多态和变形技术的原理分析

X'CON 2003

– 多态和变形病毒起源于加密解密思想，最简单的想法是对病毒进行加密，然后运行时进行解密，下面是一个简单的经过加密的病毒框架：

•**2.1.1** 多态引擎

Copyright © Hume 2003

简单加密病毒框架

```
MOV ECX , VIRUS_SIZE
MOV EDI, offset EncrptStart

DecrptLoop:
    XOR byte ptr [EDI]，key
    INC    EDI
    LOOP    DecrptLoop
```

VIRUS_SIZE是加密代码的长度。

offset EncrptStart是加密代码的起始地址。

key是密钥。

Copyright © Hume 2003

思路

- 这样加密后的病毒体只要key不同，加密病毒的特征值就消失了。但问题是解密头还是固定的，还是有特征值，这和普通的病毒相比没什么改善。改善的方法就是使每次生成的解密头和密钥都不同。应该达到如下目标：
  - 每条解密指令都不是固定的，我们看到的上面的固定代码实际上只是一种可能，病毒每次复制自身的时候，这些代码都会随机改变。
  - 密钥每次都随机生成。

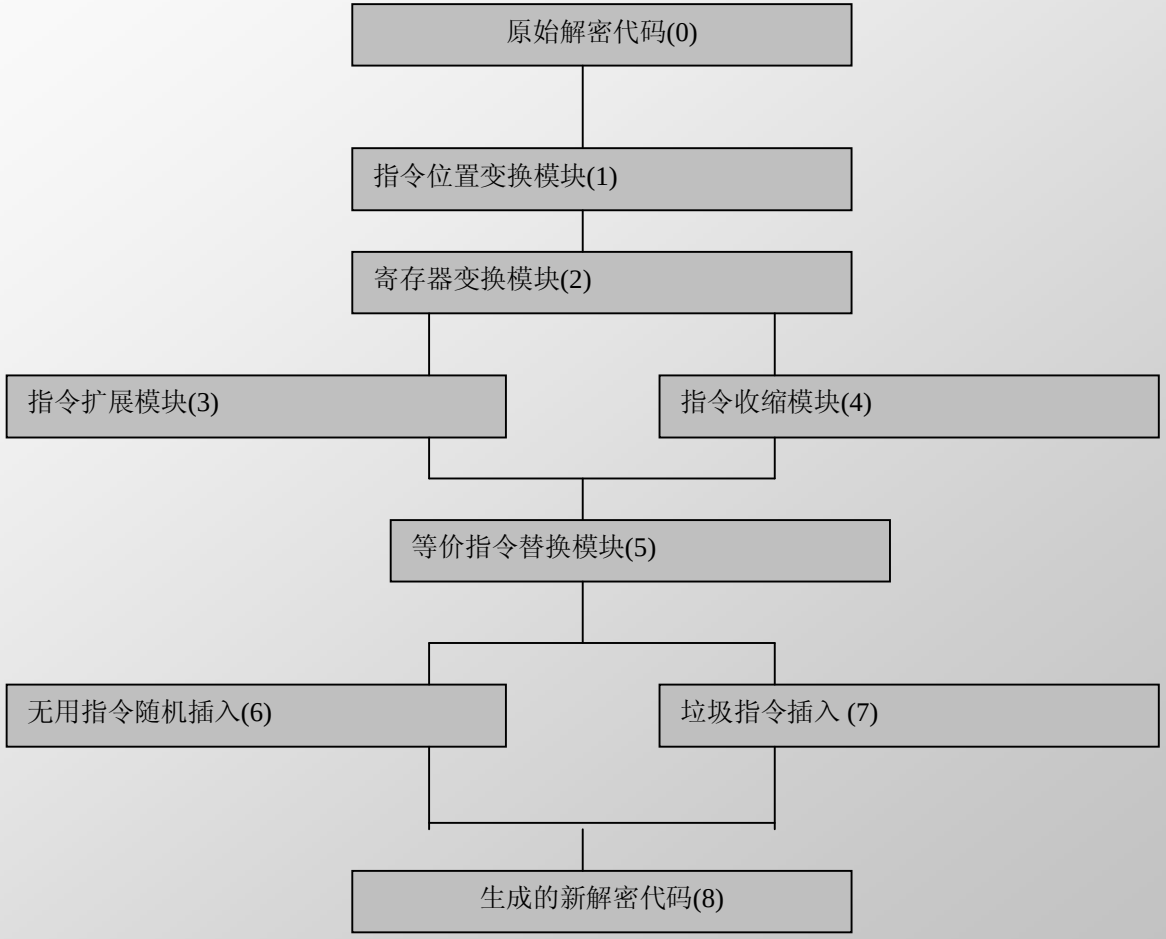
Copyright © Hume 2003

X'CON 2003

- 这样在每次感染后，因为解密代码几乎没有规律可寻，而其他代码又经过加密，所以整个病毒代码都不是固定的，如果指望能够从这些代码中找到固定的病毒特征码则是徒劳的。这就是多态病毒思想。

Copyright © Hume 2003

多态引擎的组成



Copyright © Hume 2003

X'CON 2003

- 1）的作用是变换不影响执行效果的指令的相对位置，如下指令的位置就是任意的，可以有3!=6种变化：
- mov ebx,23
- xor ecx,ecx
- lodsd

Copyright © Hume 2003

- 2）的作用是随机选取寄存器如：
  - mov reg,[123456]
  - mov [45678],reg
  - reg就可以在eax,ebx,ecx....等通用寄存器之间进行随机选择。

Copyright © Hume 2003

- 3）将一条指令替换为多条等价指令：

```
STOSD
MOV  EAX,EDX
POP   EAX
```

→

```
MOV  [EDI],EAX，ADD EDI,4
PUSH  EDX , POP   EAX
MOV  EAX,[ESP]  ADD  ESP,4
```

Copyright © Hume 2003

X'CON 2003

- 4）多条指令替换为一条等价指令：

```
MOV     [EDI],EAX
ADD     EDI,4
```

```
STOSD
```

```
PUSH    EDX
XCHG    EAX,EDX
POP     EDX
```

```
MOV     EAX,EDX
```

Copyright © Hume 2003

- 5）一条指令替换为一条等价指令：

```
XOR   EAX,EAX      SUB  EAX,EAX
ADD   EXX,1         INC   EXX
```

6）可以是不影响解密代码的指令
7）垃圾指令，不能影响代码执行效果的指令，可以有单字节、双字节等垃圾指令。

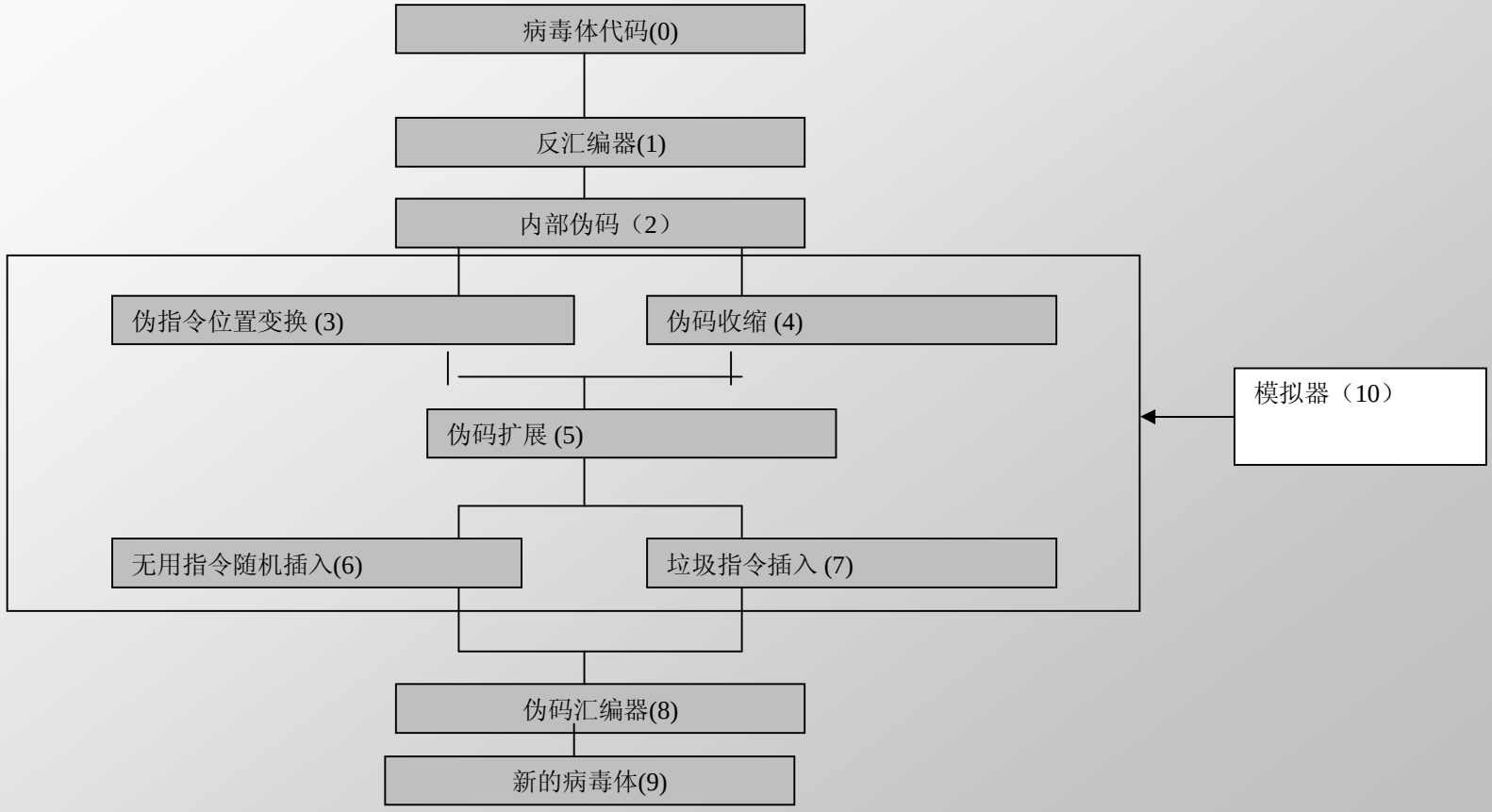
6和7的目的是为了干扰杀毒引擎扫描器。

Copyright © Hume 2003

- 以上只是一个简单的模型，一个好的解密代码生成引擎应该达到：
  - 选取的指令覆盖整个IA-32指令集
  - 解密指令选取具有充分的随机性
  - 密钥的选取具有充分的随机性
  - 加密解密算法可变

Copyright © Hume 2003

2.1.2 变形引擎



Copyright © Hume 2003

变形引擎的模块说明

- 1）反汇编器，对病毒二进制代码进行反汇编，得到中间语言。
- 3）4）5）6）7）的作用和多态引擎是类似的，区别在于后者要对伪码进行变换。首先进行伪码收缩的意图是防止在不断的传染扩展过程中产生“肥胖”的病毒体。
- 8）对变形后的伪码进行汇编，使之生成可实际运行的病毒代码。
- 10）模拟器的作用对指令的执行效果进行模拟，以更好地服务于代码的变换。实现难度很大，可选。

Copyright © Hume 2003

变形病毒的缺陷

- 1）编写难度大，调试起来困难。上述可知编写一个变形引擎需要很多知识和技巧。这使得很少人再去做这种极度痛苦的工作。
- 2）体积大。目标也大，容易引起人的怀疑。举一个例子600K的一个软件增加2K不会引起人的什么怀疑，如果600K的软件增加100K那就很可疑了。
- 3）一些设计上的漏洞，可能会被反病毒软件利用，会导致无法达到预期的效果。
- 因此对于成本综合测量通常的结论是一般的多态现阶段即可达到目标，何必再费劲去研究和编写变形病毒呢？

Copyright © Hume 2003

2.2 多态和变形技术在网络攻击中的应用

- 蠕虫主动攻击是利用多态和变形技术的好地方，他们本身就很大，也就不在乎多一点能提高攻击效果的多态和变形代码了。
- ShellCode的多态化处理躲过部分IDS检测

Copyright © Hume 2003

三、觉醒的人们——如何对付多态和变形技术

- **3.1 反病毒技术的发展**
  - 那么Aver们在做什么呢？在病毒不断进化的同时杀毒商们也没闲着，除了不着边际地吹嘘之外，也有一些技术确实极大地改变了一边病毒压倒一切的一边倒的形势。这些技术包括：

Copyright © Hume 2003

防毒卡技术

- 这种技术企图以固定的模式对付进化的软件和病毒世界结果失败了，看来只有历史研究价值了。

Copyright © Hume 2003

特征码提取和检测

- 传统的反病毒技术，提取病毒体代码的固定的特征值。到现在为止仍然发挥着重要的作用。介绍很多，不再赘述。

Copyright © Hume 2003

防患于未然——防火墙技术

X'CON 2003

- 防火墙是“文件系统实时监视技术”的通俗较法，其中心思想是利用对底层驱动等对文件读写进行过滤和扫描，试图发现任何病毒类型的行为而给出警报，防患于未然。这比等病毒感染了全部硬盘文件之后再杀除效果无疑好的多，是主动病毒防护思想的产物。所谓的主动内核技术（Active K）技术不过是防火墙思想的翻版，差别在于主动内核技术要把防火墙从应用软件级集成进操作系统的底层组件之中。这也许对微软还有些诱惑力，不过既然gates没这么干，想必还是有其不成熟的地方。

Copyright © Hume 2003

虚拟机技术和启发扫描

- 杀毒引擎中的虚拟机技术和这种技术是类似的，就是仿真某种CPU（比如X86）然后用解释的方法执行病毒体，这样病毒体不会对用户机造成任何危害，但杀毒引擎可以通过虚拟执行等待多态解密代码将病毒体解密然后再利用特征码扫描病毒体从而发现病毒。再完美的多态在完美的虚拟机面前也无济于事，因为多态病毒总要还原一个不变的“核心”。

Copyright © Hume 2003

启发扫描的行为分析

- 病毒毕竟是病毒很多行为和普通软件还是有很大区别的。比如要对可执行文件进行读写、复制自身、hookAPI、自身进行SMC、反跟踪等等，要想辨别一种未知的变形病毒只能靠这种行为分析了，根据是否具有多种可疑行为以及可疑行为的严重程度进行评估，如果严重程度超过某个阈值就报警。这种行为分析技术一般就被称为启发式查毒技术。

Copyright © Hume 2003

3.2 入侵检测技术

- 一种思路就是利用神经网络的方法通过学习以对付经过多态技术处理的未知数据包，甚至采用其它的统计和行为分析等人工智能技术进行处理。

Copyright © Hume 2003

四、

- 未来的世界—人工智能?

Copyright © Hume 2003

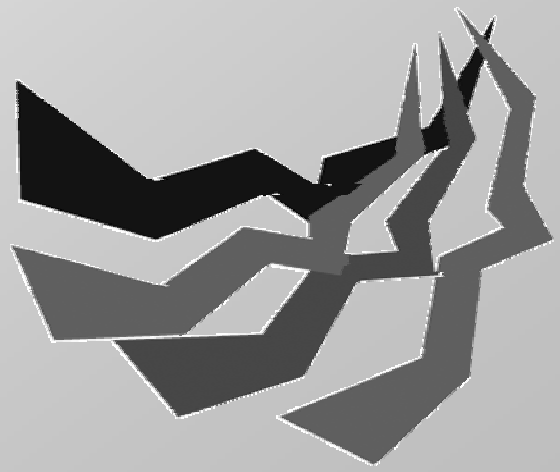
五、最后

- 软件技术发展的现状决定了现在的软件复杂度和水平，也决定了病毒和反病毒技术的水平。上面的这种“人工智能幻想”也许永远不会实现。所以，该醒醒了.....
- 但作为安全工作研究者，我们还是要关注最新的病毒和反病毒技术的进展。

Copyright © Hume 2003

谢谢！

Thanks !



X'CON 2003