网络蠕虫的工作原理及预警技术研究

北京大学 软件与微电子学院 文伟平
中科院软件研究所 蒋建春

网络蠕虫的定义 功能结构及工作机制

网络蠕虫的定义

早期恶意代码的主要形式是计算机病毒。1988年“Morris”蠕虫爆发后，Spafford为了区分蠕虫和病毒，对病毒重新进行了定义，他认为，“计算机病毒是一段代码，能把自身加到其它程序包括操作系统上；它不能独立运行，需要由它的宿主程序运行来激活它”。而网络蠕虫强调自身的主动性和独立性。Kienzle和Elder从破坏性、网络传播、主动攻击和独立性四个方面对网络蠕虫进行了定义：网络蠕虫是通过网络传播，无需用户干预能够独立地或者依赖文件共享主动攻击的恶意代码。根据传播策略，他们把网络蠕虫分为三类：Email蠕虫、文件共享蠕虫和传统蠕虫。《Internet蠕虫研究》一文中认为蠕虫具有主动攻击、行踪隐蔽、利用漏洞、造成网络拥塞、降低系统性能、产生安全隐患、反复性和破坏性等特征，并给出相应的定义：“网络蠕虫是无须计算机使用者干预即可运行的独立程序，它通过不停的获得网络中存在漏洞的计算机上的部分或全部控制权来进行传播。”该定义包含了Kienzle和Elder定义的后两类蠕虫，不包括Email蠕虫。2003年10月的世界蠕虫会议上，Schechter和Michael D. Smith提出了一类新型网络蠕虫，Access For Sale蠕虫，这类蠕虫除上述定义的特征之外，还具备身份认证的特征。

综合上述分析，本文认为“网络蠕虫是一种智能化、自动化，综合网络攻击、密码学和计算机病毒技术，无需计算机使用者干预即可运行的攻击程序或代码，它会扫描和攻击网络上存在系统漏洞的节点主机，通过局域网或者国际互联网从一个节点传播到另外一个节点”。该定义体现了新一代网络蠕虫智能化、自动化和高技术化的特征。

网络蠕虫的功能结构

Jose Nazario等人提出了蠕虫的一个功能结构框架，把蠕虫的功能模块分为六个部分：搜索模块(Reconnaissance Capabilities)、特殊攻击模块(Specific Attack Capabilities)、命令操作界面模块(A Command Interface)、通信模块(Communications Capabilities)、智能模块(Intelligence Capabilities)和非攻击使用模块(Unused Attack Capabilities)。该框架主要是对未来蠕虫的预测，难以准确地表达当前网络蠕虫的功能结构。在上述基础上本文归纳认为，网络蠕虫的功能模块可以分为主体功能模块和辅助功能模块。实现了主体功能模块的蠕虫能够完成复制传播流程，而包含辅助功能模块的蠕虫程序则具有更强的生存能力和破坏能力。

1.主体功能模块

主体功能模块由四个模块构成：①信息搜集模块。该模块决定采用何种搜索算法对本地或者目标网络进行信息搜集，内容包括本机系统信息、用户信息、邮件列表、对本机的信任或授权的主机、本机所处网络的拓扑结构，边界路由信息等等，这些信息可以单独使用或被其他个体共享；②扫描探测模块。完成对特定主机的脆弱性检测，决定采用何种的攻击渗透方式；③攻击渗透模块。该模块利用②获得的安全漏洞，建立传播途径，该模块在攻击方法上是开放的、可扩充的；④自我推进模块。该模块可以采用各种形式生成各种形态的蠕虫副本，在不同主机间完成蠕虫副本传递。例如“Nimda”会生成多种文件格式和名称的蠕虫副本；“W32.Nachi.Worm”利用

系统程序（例如TFTP）来完成推进模块的功能等等。

2.辅助功能模块

辅助功能模块是对除主体功能模块外的其他模块的归纳或预测，主要由五个功能模块构成：①实体隐藏模块。包括对蠕虫各个实体组成部分的隐藏、变形、加密以及进程的隐藏，主要提高蠕虫的生存能力；②宿主破坏模块。该模块用于摧毁或破坏被感染主机，破坏网络正常运行，在被感染主机上留下后门等；③信息通信模块。该模块能使蠕虫间、蠕虫同黑客之间能进行交流，这是未来蠕虫发展的重点；利用通信模块，蠕虫间可以共享某些信息，使蠕虫的编写者更好地控制蠕虫行为；④远程控制模块。控制模块的功能是调整蠕虫行为，控制被感染主机，执行蠕虫编写者下达的指令；⑤自动升级模块。该模块可以使蠕虫编写者随时更新其它模块的功能，从而实现不同的攻击目的。

3.网络蠕虫的工作机制

网络蠕虫的攻击行为可以分为4个阶段：信息收集、扫描探测、攻击渗透和自我推进。信息收集主要完成对本地和目标节点主机的信息汇集；扫描探测主要完成对具体目标主机服务漏洞的检测；攻击渗透利用已发现的服务漏洞实施攻击；自我推进完成对目标节点的感染。

相关蠕虫防范技术分析

网络蠕虫已经成为网络系统的极大威胁，由于网络蠕虫具有相当的复杂性和行为不确定性，网络蠕虫的防范需要多种技术综合应用，包括网络蠕虫监测与预警、网络蠕虫传播抑制、网络蠕虫漏洞自动修复、网络蠕虫阻断等，本文下面将主要讨论近几年的网络蠕虫检测防御技术。

基于GrIDS的网络蠕虫预警

著名的GrIDS主要针对大规模网络攻击和自动化入侵设计的，它收集计算机和网络活动的数据以及它们之间的连接，在预先定义的模式库的驱动下，将这些数据构建成网络活动行为来表征网络活动结构上的因果关系。它通过建立和分析节点间的行为图（Activity Graph），通过与预定义的行为模式图进行匹配，检测网络蠕虫是否存在，是当前检测分布式网络蠕虫入侵有效的工具。

但是通过分析认为，GrIDS在检测网络蠕虫方面仍存在以下不足：GrIDS的探测点对网络中传输的包信息不进行基于上下文的相关性分析，没有充分利用更多的、有效的数据，只作简单的基于事件的关联分析；GrIDS没有对TCP连接中的目标地址和目标服务作有效性分析，而上述分析是判断未知网络蠕虫入侵网络的重要依据；GrIDS检测到网络蠕虫后，由于没有建立任何响应机制，不能提供与内部探测点和外部防火墙的互动，因此不能形成有效的预警和防范机制。针对上述GrIDS的弱点，本文作者已设计了一种基于网状关联分析预警网络蠕虫攻击的新方法，它采用分布式体系结构，充分利用网络环境中各探测点提供的信息和数据，采用数据挖掘和异常检测的思想，通过对各探测点之间的数据作关联分析，基本实现了大规模网络环境下分布式网络蠕虫入侵的预警。

基于PLD硬件的蠕虫预警

华盛顿大学应用研究室的John W. Lockwood、James Moscola1和Matthew Kulig等提出了一种采用可编程逻辑设备（Programmable Logic Devices，PLDs）对抗网络蠕虫的防范系统。该系统由三个相互内联部件DED（Data Enabling Device）、CMS（Content Matching Server）和RTP（Regional Transaction Processor）组成。DED负责捕获流经网络出入口的所有数据包，根据CMS提供的特征串或规则表达式对数据包进行扫描匹配并把结果传递给RTP；CMS负责从后台的MYSQL数据库中读取已经存在的蠕虫特征，编译综合成DED设备可以利用特征串或规则表达式；RTP根据匹配结果决定DED采取何种操作。网络蠕虫大规模入侵时，系统管理员首先把该蠕虫的特征添加到CMS的特征数据库中，DED扫描到相应特征才会请求RTP做出放行还是阻断等响应。

系统具有以下优点：①DED采用高速硬件FPX（Field-programmable Port Extender）实现其核心功能，对数据包的扫描速率可以实现2.4Gbps，所以该系统能够实现大规模高速网络环境对网络蠕虫的检测；②高速硬件FPX比软件系统更容易实现并行技术。

系统存在的不足：①只能进行事后处理，不能检测和防御未知蠕虫；②采用特征匹配技术，存在一定的误警率。

基于HoneyPot的蠕虫预警

早期HoneyPot主要用于防范网络黑客攻击。ReVirt是能够检测网络攻击或网络异常行为的HoneyPot系统。Spitzner首次运用HoneyPot防御恶意代码攻击。相关文献提出了采用虚拟HoneyPot检测和阻断网络蠕虫攻击的防范框架，其主要实现是在边界网关或易受到蠕虫攻击的地方置放多个的虚拟HoneyPot，HoneyPot之间可以相互共享捕获的数据信息，采用NIDS的规则生成器产生网络蠕虫的匹配规则，当网络蠕虫根据一定的扫描策略扫描存在漏洞主机的地址空间时，HoneyPots可

以捕获网络蠕虫扫描攻击的数据，然后采用特征匹配来判断是否有网络蠕虫攻击。此外HoneyPot能够阻断网络蠕虫的攻击。Oudot采用HoneyPot实现对“W32.Blaster”的检测与防御。

HoneyPot主要具有以下优点：①HoneyPot可以转移蠕虫的攻击目标，降低蠕虫的攻击效果；②HoneyPot为网络安全人员研究蠕虫的工作机制、追踪蠕虫攻击源和预测蠕虫的攻击目标等提供了大量有效的数据；③由于网络蠕虫缺乏判断目标系统用途的能力，所以HoneyPot具有良好的隐蔽性。

HoneyPot存在以下一些不足：①HoneyPot能否诱骗网络蠕虫依赖于大量的因素，包括HoneyPot命名、HoneyPot置放在网络中位置和HoneyPot本身的可靠性等；②HoneyPot可以发现大量扫描行为（随机性扫描、顺序扫描等）的网络蠕虫，但针对路由扫描和DNS扫描蠕虫时，效果欠佳；③HoneyPot很少能在蠕虫传播的初期发挥作用。

良性蠕虫抑制恶意蠕虫

最早网络蠕虫引入计算机领域就是为了进行科学辅助计算和大规模网络的性能测试，蠕虫本身也体现了分布式计算的特点，所以可以利用良性蠕虫来抑制恶意蠕虫。良性蠕虫首先应具有高度的可控性和非破坏性，其次尽量避免增加网络负载。良性蠕虫可以采用几种传播方式：①利用恶意蠕虫留下的后门；②利用恶意蠕虫攻击的漏洞；③利用其他未公开的系统漏洞；④利用被攻击主机的授权。良性蠕虫可以有效的消除恶意蠕虫，修补系统漏洞，从而减少网络中易感主机的数量。“Cheese”蠕虫利用“Lion”蠕虫留下的后门控制被感染的主机，清理掉主机上的“Lion”蠕虫留下的后门，修补系统的漏洞。针对“CodeRed”的对抗蠕虫“CRClean”的代码也曾经被公布过，但最后它们没有实际的被释放到网络中。“W32.Nachi.Worm”利用“W32.Blaster”的使用的系统漏洞对抗“W32.Blaster”。上述例子都是蠕虫对抗蠕虫的经典实例。“Cheese”、和“W32.Nachi.Worm”都不是良性蠕虫，因为它们对网络负载造成严重影响。

良性蠕虫具有以下优势：①良性蠕虫对用户透明，不需要隐蔽模块，可以充分利用集中控制的优势，主体程序、数据和传播目标都从控制中心获得；②采用分时分段慢速传播，尽量不占用网络带宽和主机资源；③同一个良性蠕虫可以执行不同的任务，只需从控制中心下载不同的任务模块，包括进行分布式计算或者采集网络数据等等，然后将结果汇总到控制中心。

良性蠕虫是未来蠕虫研究的方向，其设计的关键在于可控性设计，因此设计良性蠕虫要考虑更多的不可确定性因素，尚需进一步深入研究。

基于CCDC的蠕虫检测、防御和阻断

由于网络蠕虫具有生物病毒特征，美国安全专家提议建立CCDC（The Cyber Centers for Disease Control）来对抗网络蠕虫攻击。防范网络蠕虫的CCDC体系实现以下功能：①鉴别蠕虫的爆发期；②蠕虫样本特征分析；③蠕虫传染对抗；④蠕虫新的传染途径预测；⑤前摄性蠕虫对抗工具研究；⑥对抗未来蠕虫的威胁。CCDC能够实现对大规模网络蠕虫入侵的预警、防御和阻断。但CCDC也存在一些问题：①CCDC是一个规模庞大的防范体系，要考虑体系运转的代价；②由于CCDC体系的开放性，CCDC自身的安全问题不容忽视；③在CCDC防范体系中，攻击者能够监测蠕虫攻击的全过程，深入理解CCDC防范蠕虫的工作机制，因此可能导致突破CCDC防范体系的蠕虫出现。

其它

除了上述技术以外，网络蠕虫防范技术还有很多。目前比较流行的抑制网络蠕虫传播的方法就是在路由节点屏蔽和过滤含有某个网络蠕虫特征的报文。此外，邹长春等提出，通过对一定地址空间的流量监控来预测网络蠕虫的传播，从而采取更有效的措施来对抗网络蠕虫的大规模攻击。由Liston所设计的LaBrea工具，能够通过长时间阻断与被感染机器的TCP连接来降低网络蠕虫的传播速度。

基于网状关联的蠕虫预警技术研究

网络蠕虫预警机制的原理性探索

网络蠕虫是一种智能化、自动化的攻击载体，它会扫描和探测网络上存在服务漏洞的节点主机，一旦渗透成功，会自我复制许多副本，通过局域网、国际互联网或者电子邮件从一个节点传播到另外一个节点。

网络蠕虫的攻击行为可以分为4个阶段：信息收集、弱点探测、攻击渗透和自我复制。信息收集主要完成对目标网络和主机的信息汇集，包括目标网络拓扑结构和网络中节点主机的操作系统类型；弱点探测主要完成对具体目标主机服务漏洞的检测；攻击渗透利用已发现的服务漏洞实施攻击；自我复制完成对目标节点的感染。网络蠕虫在整个攻击过程中，要向目标网络和目的节点发送大量的服务请求。

网络蠕虫传播具有以下特征：

(1) 传出数据的相似性。网络蠕虫在传播的各个阶段，感染节点传出的数据具有相似性。数据包都包含了相应的请求、攻击代码或蠕虫的主体程序，内容相对稳定，因此其传输数据

的大小基本不变。例如Code Red利用IIS漏洞进入被感染节点的主机后，产生一百个线程，前面九十九个线程都利用随机产生的IP地址，探测其它节点主机是否存在IIS的Indexing Service缓冲区溢出漏洞。在一定时间内，每个线程对目的主机的请求内容都是相似的 。

(2) 大量的无效IP地址和无效服务请求：网络蠕虫为了在网络中迅速传播和扩散，攻击目标的选择具有盲目性。信息的收集和探测都会导致大量无效IP地址的产生，由于攻击目标IP地址的无效性，因此相应的服务请求也得不到应答。

(3) 节点间的传播行为具有相似性：网络蠕虫感染一个节点主机后，这个节点成为新的蠕虫载体，并开始扫描、探测、攻击新的目标节点，这个传播行为和最初发生的传播行为具有相似性。

从上面对网络蠕虫攻击行为模式的分析可以看出，如果网络中的某一节点主机在短时间内对外进行大量的TCP连接请求，这些连接请求的目标端口相同、数据包大小一定、数据包内容相似，且目标IP地址和目标服务都得不到应答，则可判断该节点可能被某种蠕虫侵入。在一定时间范围内，如果网络中另外一个节点也发生了与上述节点类似的行为，则可判断某种网络蠕虫已经侵入该网络并正在扩散。这时，控制中心应当立即预警，并采取有效措施阻止网络蠕虫的大规模探测、渗透和自我复制。这就是通过网状关联分析预警网络蠕虫的基本思想。

基于网状关联分析的网络蠕虫预警

网络蠕虫对新的目标节点主机进行扫描、自我复制和传播，在网络上传播的路径以网状的方式呈现。如图1所示，网络蠕虫从节点A传播至节点B、节点C、节点D、节点E和节点F，由于节点F受到网络蠕虫感染，所以网络蠕虫再由节点F传播至节点G与节点H。长期来看，网络蠕虫传播扩散的路径会产生一个较大的网状图形，如图2所示。

网络数据的传输行为，可以通过数据流的源节点与目标节点的关联分析，描绘成网状图。以下，我们通过探测点或控制中心分析数据传输行为，对数据流信息作关联分析。对异常网络的数据传输行为，进行较长时间的观察。在本节的以下部分，我们给出若干定义和预警算法。首先，我们定义网络蠕虫传播的数据传输行为，其中，节点A和节点B是控制中心和所有探测点中的任意两个节点。

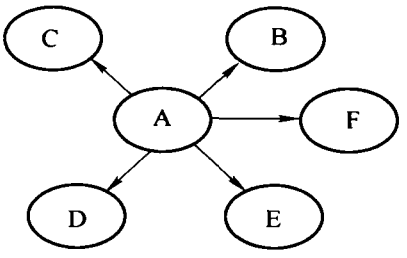

图1 网络蠕虫传播路径图

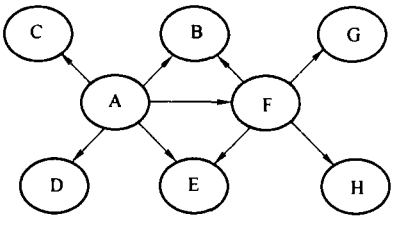

图2 网络蠕虫扩散图

定义1：数据传输行为 在TCP协议中，四元组（srcHost,srcPort,dstHost,dstPort）惟一确定一个TCP连接(srcHost为源主机IP，srcPort为源端口，dstHost为目的主机IP，dstPort为目的端口)。在本文中，为了对数据流进行更细致的分析，对此定义进行扩充，定义如下七元组（timestamp，srcHost，srcPort，dstHost，dstPort，dataSize，data）为唯一确定的TCP连接。其含义是，在时间timestamp源节点主机（srcHost，srcPort）向目的节点主机（dstHost，dstPort）发出TCP连接请求，dataSize为此次连接请求交换数据的大小，data为数据的前m个字节，其中，m为可配置参数。

定义2：数据传输行为集合 所有源节点主机（srcHost，srcPort）向目的节点主机（dstHost，dstPort）发出的TCP连接请求的集合C，Ci表示C中第i个TCP连接（1□i□n，n为集合C中所含TCP连接的个数）。

定义3：传出信息属性 对于任意一个TCP连接Ci（Ci ∈ C），定义ATTR_FIRST（timestamp，srcHost，dstPort，dataSize，noHost，noService）为此连接的第一传出信息属性，记为ATTR_FIRST（Ci），其中noHost是dstHost有效性判断谓词，noService是请求服务有效性谓词。当dstHost没有应答时，noHost之值为0；反之为1。当目的端口的服务请求没有应答时，noService之值为0，反之为1。定义ATTR_SECOND（dstPort，dataSize，data，noHost，noService）为此连接的第二传出信息属性，记为ATTR_SECOND（Ci）。第一传出信息属性主要用于判断一个节点是否存在异常数据传出行为。当存在异常数据传出行为的节点时，第二传出信息属性主要用于这些节点间的相似度计算。

定义4：传出信息属性一致 对于任意两个TCP连接Ci和Cj(Ci,Cj ∈ C)，如果ATTR_FIRST(Ci) = ATTR_FIRST(Cj)，则称这两个连接的传出信息属性一致。以节点主机(srcHost)为源节点的m个TCP连接Ci，Ci+1，Ci+m-1，若∀

k ∈[1,m-1]，ATTR_FIRST(Ci)=ATTR_FIRST(Ci+k)，则称节点主机srcHost存在传出信息属性一致的m个连接 。

定义5：节点异常数据传出行为 如果节点A在特定的时间范围t内，存在传出信息属性一致的N个连接（N为判定节点异常数据传出行为的阈值），则认为节点A发生了异常数据传出行为。记与节点A异常数据传出行为相关的TCP连接集合为[CA]={CA 1,C A 2,…,C A i,…,C A N}。N为[CA]中元素的个数，记作|[CA]|=N，其中CA为该集合的特征连接。

定义6：节点间异常数据传出行为间的相似性 如果节点A和节点B发生了异常数据传出行为，且与节点A和节点B的异常传出行为相关的TCP连接集合[CA]和[CB]中的第二传出信息属性ATTR_SECOND(CA)与ATTR_SECOND(CB)的相似度大于设定阈值时，则认为节点A和节点B发生的异常传出行为是相似的。

当节点A和节点B发生的异常数据传出行为相似时，则可判断节点A和节点B已被同类型网络蠕虫感染。

根据第2节对网络蠕虫预警原理的分析，结合本节给出的定义，我们给出如下的网络蠕虫预警算法。预警算法有2个：探测点的节点异常传出行为分析算法、控制中心数据关联分析算法，分别描述如下：N：判断节点异常数据传出行为的阈值；T：时间阈值；S：相似度阈值；A,B：控制中心和所有探测点中的两个节点；ABNORMAL：存储送往控制中心的，与异常数据传出行为相关的TCP连接集合的缓冲池；WN：蠕虫网络；Sim(ATTR_SECOND(CA),ATTR_SECOND(CB))：传出信息属性相似度计算函数。

探测点分析节点异常传出行为的算法如下：

```
/* 判断节点的异常传出行为,A为预警网络中具有唯一标识的节点 */
Begin
Create Profile A ATTR_FIRST(CA)=( IPA, dstPortA, dataSizeA,0,0)
for each Connection C i: < timestamp,srcHost,srcPort, dstHost , ···>
create ATTR_FIRST (C i)=(srcHost, dstPort,dataSize, noHost,noService)
if ATTR_FIRST (Ci)∩ ATTR_FIRST(CA) = ATTR_FIRST(CA)
Then
save C i to [CA]
End if
If |[CA]| □N and (timestampAN - timestamp A N-N+1)<T
Then
Send [CA] to ABNORMAL
Empty [CA]
go to Begin
End if
End for
End begin
```

控制中心对异常传出行为进行关联分析的算法如下：

```
begin
Receive: [CA]←ABNORMAL
Receive: [CB]←ABNORMAL
If Sim(ATTR_SECOND(CA),ATTR_SECOND(CB))>S
Then
A,B ∈ WN
Send (warning ,Response) to all sensors
End if
End begin
```

上述算法首先由探测点获得Connection C，提取与节点A相关的TCP连接，并建立第一传出信息属性集合ATTR_FIRST。当节点A在一定的时间T内，存在N次传出信息属性一致的连接请求，则可判断节点A有异常数据传出行为。同样可以判断，节点B是否存在异常数据传出行为。当异常传出行为集合ABNORMAL的元素有两个或两个以上时，便可对其属性进行相似度计算：Sim(ATTR_SECOND(CA),ATTR_SECOND(CB))。若相似度大于我们规定的阈值S，则判断节点A和节点B已经加入某种类型的蠕虫网络WN，并向所有的探测点发送预警响应指令。

网络蠕虫预警系统的结构模型

网络蠕虫检测的目的在于发现网络中的节点主机是否感染网络蠕虫，而网络蠕虫预警的主要功能是在网络蠕虫尤其是未知的网络蠕虫大规模探测、渗透和自我复制之前，及时发现痕迹进行预警，并采取相应的有效措施。本系统采用分布式协同预警体系结构模型，该模型的基本策略是将大规模网络划分为若干子网，在每个子网中设立探测点，同时建立一个控制中心，在各个子网的探测点和控制中心之间建立预警通道。一旦局部发现疫情，通过预警通报机制能够迅速将该警报信息通知控制中心及其它探测点，从而形成全网络的协同预防机制 。

在网络蠕虫预警模型中，探测点能够完成数据采集、节点异常检测和预警响应功能。控制中心完成数据关联分析、抽取蠕虫特征样本、向探测点发送预警响应指令和分发最新蠕虫特征样本的功能。所以要求探测点和控制中心之间能够很好地协同工作，具体来说主要有以下两个方面：①数据分析协同。探测点对自己采集到的数据进行模式匹配分析，发现一些已知的蠕虫攻击行为；同时，采用异常检测技术判断本网段各节点的异常数据传输行为，并把相关异常数据上报给控制中心。控制中心根据各探测点上报的异常数据进行网状关联分析。②预警响应协同。控制中心如果发觉节点存在异常，立即把预警响应指令发送给异常的探测点；同时，挖掘未知蠕虫，抽取蠕虫特征样本，并向各探测点分发最新蠕虫的攻击特征码。探测点接收控制中心的预警响应指令和最新蠕虫的攻击特征码，及时更新本地的蠕虫攻击特征样本库，并把响应结果返回控制中心。

（责编 褚德坤）