第 33 卷 第 5 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.5 Computer Engineering March 2007
·安全技术· 文章编号：1000—3428(2007)05—0132—03 文献标识码：A 中图分类号：TP309

Honeynet 技术研究与实例配置

冯朝辉[1]，范锐军[2]，张 彤[1]
(1. 西北核技术研究所，西安 710024；2. 西北工业大学，西安 710072)

摘 要：Honeynet 是一种高互动蜜罐，其目的是搜集有关安全威胁的全面信息。Honeynet 是一种体系结构，其首要需求是数据控制和数据捕获。该文构建了一个完整的 Honeynet 实例，论述了其中的关键技术，分析了 Honeynet 的特有风险。
关键词：Honeynet；蜜罐；数据控制；数据捕获

Technology Research and Building Example of Honeynet

FENG Zhaohui[1], FAN Ruijun[2], ZHANG Tong[1]
(1. Northwest Institute of Nuclear Technology, Xi'an 710024; 2. Northwestern Polytechnical University, Xi'an 710072)

【Abstract】 Honeynet is a form high-interaction honeypot. Its aim is to gather extensive information on threats. A honeynet is an architecture, the two critical requirements for this architecture are data control and data capture. The paper discusses how to build a honeynet, and describes the key techniques. It analyzes the unique risks with honeynet.
【Key words】 Honeynet; Honeypot; Data control; Data capture

1 Honeynet 技术

传统上，信息安全技术完全是防御性的，如防火墙、入侵检测、加密等，这些机制防御性地保护资源、检测防御中的漏洞，针对这些漏洞作出响应，在这种纯粹的防御策略中，敌人拥有主动权。Honeynet技术试图改变这种状况，Honeynet的目的是搜集安全威胁的信息，以发现新的入侵工具、测定攻击模式、研究攻击者的动机，利用Honeynet搜集的信息可以更好地理解和对付来自内部和外部的威胁[1,2]。

1.1 Honeynet 的概念

Honeynet 是蜜罐(honeypot)的一种，一个蜜罐就是一个设计用来观测黑客如何探测并最终入侵系统的一个系统，这种系统包含一些并不威胁公司的数据或应用程序，同时对于黑客来说又具有很大的诱惑力，如放置在网络上的一台计算机，表面看来像一台普通的机器但同时通过一些特殊配置来引诱潜在的黑客并捕获他们的踪迹，就像捕鼠器一样。

Honeynet属于高互动(high-interaction)蜜罐，高互动意味Honeynet提供真实的系统、应用和服务与攻击者交互，从而获得有关一个组织内、外部威胁的全面信息[2,3]。Honeynet是一个包含一个或多个Honeypot的网络，是一个完整的网络系统，其内的资源(即Honeypot)可以是设置者想要提供的任何类型的系统、服务或信息，例如Solaris服务器上的Oracle数据库、Windows系统上用IIS发布的一个电子商务站点等。

由于Honeypot不是营运系统(production system)，因此Honeynet本身也没有营运活动，即没有授权的服务[2,3]。任何与Honeynet的互动都是恶意的或非授权的，任何进入Honeynet的连接都是探测、扫描或攻击，任何从Honeynet发出的连接就表示已经有人闯入系统，并向外发起活动。所有从Honeynet捕获的信息都是与攻击有关的，具有低噪声的特点，这一点使得分析Honeynet内的活动比较简单[2,3]。而传统的安全技术，如防火墙日志或IDS检测信息，既包含攻击信息也包含合法的系统活动信息，因此必须耗费大量的时间、精力从数千条报警中筛选关键信息，以识别攻击。

Honeynet不是一个产品，不是安装在计算机上的一个软件，而是一种体系结构，这种结构构建一个高度受控的网络，网络内的所有活动都受到控制和监视，该网络容忍入侵，用于分析入侵行为[2]。

1.2 Honeynet 体系结构需求

Honeynet的实现难点在于配置，若配置失误，则可能无法捕获攻击者的行为，还可能把其它Honeynet以外的系统暴露在更大的风险之下。成功地配置Honeynet有两个关键需求：数据控制和数据捕获[2,3]。数据控制在攻击者不知道的情形下，限定了攻击者与Honeynet的交互活动，即规定攻击者可以做什么、不可以做什么。数据捕获是在攻击者不知情的情况下，捕获攻击者在Honeynet内的所有活动。

数据控制对攻击者的活动进行限制，以降低风险，这里的风险是指攻击者利用 Honeynet 去攻击、危害 Honeynet 以外的系统。要做到这一点有一定难度：必须给予攻击者一定的行动自由，攻击者获得的行动自由度越大，能完成的活动越多，Honeynet 就能捕获更多的攻击行为；然而，攻击者获得的自由越大，他们绕过数据控制并危害其它 Honeynet 以外系统的风险就越大。Honeynet 的设置者必须根据自己的需求和所能承担的风险来决定在多大程度上限制攻击者的行为。数据控制应该采用多种机制、分层实现的方法，如计算外出连接数目、设置入侵防护网关、限制带宽等，不同机制的联合可防止单点失效，并有利于搜集新的、未知的攻击信息。

数据捕获监视和记录攻击者在 Honeynet 内的所有行为，然后通过分析捕获到的数据来研究攻击者的工具、策略和动

基金项目：国防科技预研项目
作者简介：冯朝辉(1978－)，女，研究实习员、硕士，主研方向：计算机网络安全；范锐军，博士生；张 彤，博士、研究员
收稿日期：2006-03-24 **E-mail：**fengzhh503@126.com

机。数据捕获也应该采用多种机制、在多层次上实现，这样不仅能把攻击者的行为整合在一起，还能有效防止单点失效。同时还应尽量减小攻击者探测数据捕获机制的能力，可以采用以下两种方法：首先尽可能少地更改 Honeypot，改动越多被探测到的可能性就越大；其次，不要在捕获数据的 Honeypot 上存储被捕获到的数据，捕获到的数据可能被攻击者探测到，还可能被他们修改和删除，应该把这些信息记录、存储在一个与 Honeynet 隔离的安全系统中。

第 3 个需求是数据聚集[2]，这只适用于在分布式环境下部署多个Honeynet的组织，如Honeynet Research Alliance，必须将捕获到的数据汇集并存储到一个中心数据库中，从而综合分析数据，提高数据的研究价值。

2 Honeynet 构建、配置实例

2.1 网络结构

本文所构建、配置的 Honeynet 的结构如图 1 所示。

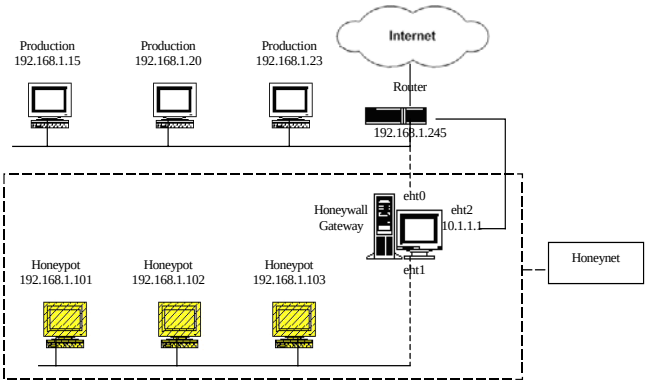

图 1 Honeynet 配置实例的结构

图 1 中Honeynet的关键组件是一个网关，由它将提供给攻击者攻击的Honeynet与其它运营系统隔离开来，相当于一堵墙，因此被称为Honeywall。所有进入或离开Honeynet的通信都必须经过Honeywall，因此Honeywall成为Honeynet的指挥和控制中心，许多功能都是在它上面构建的。网关Honeywall是一个工作在数据链路层的网桥，它的外部接口(eht0)与营运系统网络相连，内部接口(eht1)与Honeynet网络相连，由于这个网关是一个网桥，因此内外系统在同一个IP网络上，这样不仅可以追踪和了解来自外部的安全威胁，也可以监视内部威胁。网关的第 3 个接口(eth2)用于远程管理，例如把日志或捕获的数据移动到一个中心数据库中。网关的内外接口(eth0 和eth1)工作在网桥模式，因此没有IP地址，第 3 个接口(eth2)分配有IP堆栈，其IP地址为 10.1.1.1，是个独立、安全的网址，只用于管理目的。这种结构的优点在于：由于网关没有路由跳跃点、没有TTL开销、也没有与其绑定的MAC地址，因此很难被探测到；而且可以在这个单一的网关上同时实现数据控制和捕获，从而简化了Honeynet的配置[3,4]。

接下来的工作是配置网关，使其符合Honeynet体系结构需求。在本文的例子中，网关Honeywall运行在基于X86 硬件的Linux系统上，内核版本为 2.4.18-3。下一步要确保该网关支持网桥功能，大多数的Linux版本都缺省支持网桥功能，如果系统不支持，可以从网站http://bridge.sf.net/上下载网桥的rpm包或源代码。但是，Linux的大多数版本在网桥模式下不支持IPTables[5](Linux上的软件防火墙)，IPTables是很重要的，它不仅能使网关本身更加安全，还用于Honeynet中的数据控制。为了能在网桥模式下使用IPTables，必须要配置系统内核的IPTables选项并重新编译。

Honeynet Project提供了一个配置脚本rc.firewall[6]，该脚本示例如何配置网关上的功能，包括网桥、防火墙、管理接口配置、网关管理、网络活动记录等，使其符合Honeynet的要求。脚本中详细描述了每个变量的定义、作用，在运行该脚本之前，只需要按需求设置这些变量的值，在目录/bin/bash下运行该脚本。

2.2 数据控制

本文的例子使用两种技术来实现数据控制：连接限制和NIPS(网络入侵防护系统)。连接限制是限制 Honeynet 向外发出的连接数目，NIPS 的作用是阻断已知攻击。两种技术的联合使用，为数据控制提供了冗余性和灵活性。由于所有进入、外出 Honeynet 的通信都经过网关，网关成为攻击者活动的咽喉，因此应该在网关上实现数据控制。

连接限制计算从 Honeynet 向外发出的连接数目，当达到限制数时，就阻断后继的所有连接，这是减少那些需要大量外出连接的攻击活动的首要方法，如大规模的扫描、拒绝服务等。使用脚本 rc.firewall 配置 IPTables，来实现连接控制，脚本 rc.firewall 中对外出连接的限制如下所示，它设置了攻击者从 Honeynet 能够向外发出的各类连接的数目，其中变量 OTHER 是指除 TCP、UDP、ICMP 以外的 IP 协议(如 IPSec、IPv6 隧道、网络语音协议等)。

```
### Set the connection outbound limits for different protocols.
SCALE="day"
TCPRATE="15"
UDPRATE="20"
ICMPRATE="50"
OTHERRATE="15"
```

以上脚本代码说明了 IPTables 对各类外出连接数目的限制，如 TCP 连接每天最多 15 个。当攻击者侵入 Honeynet 中的一个 Honeypot 后，他们可能出于各种动机向外发出连接(如下载工具包、安装自动蠕虫、IRC 聊天、发送电子邮件等)。每次建立外出连接的时候，防火墙 IPTables 对其计数，当达到限制的数目后，IPTables 阻断所有后继的同类外出连接。

若连接限制控制 Honeynet 每天最多可以向外发出 15 个 TCP 连接，那么当 Honeynet 感染蠕虫后，被允许的 15 个外出连接将感染 Honeynet 以外的系统，所以连接限制只能减少被感染的系统数目，必须要通过其它的机制阻断攻击，这就是 NIPS。

NIPS检查每个通过Honeywall的数据包，如果发现某个数据包与IDS的一条检测规则匹配，就产生告警，并丢弃或修改这个数据包，从而阻止攻击，其功能与传统的NIDS(网络入侵检测系统)相似。NIPS可以大大减少入侵者利用Honeynet向外发出攻击的风险，但也有局限性，它只能检测到已知的攻击。在本文的实例中，使用一个修改版本的Snort[7]——Snort_inline来实现NIPS(Snort是一种NIDS，可以丢弃、修改数据包)。

Snort_inline 工作在 Honeywall 上，是在网关模式下实现 NIPS 的，因此必须有数据包路由能力，但 snort_inline 本身没有路由功能(即 ip_forward)，需要其它的组件来为 snort_inline 完成数据包的路由。路由功能由 IPTables 实现。配置 IPTables 完成以下功能：从内核堆栈获取数据包，提交给用户态下的 snort_inline 进行分析，然后取得分析后的数据包，传递给内核，继续数据包的处理。IPTables 的这种机制称为 user-space queuing，要求内核必须装载并激活 ip_queue

模块，然后要在脚本 rc.firewall 中允许“ QUEUE ”选项。

```
# IPTables script can be used with the Snort-Inline filter
QUEUE="yes"                # Use experimental QUEUE support
```

snort_inline 和 IPTables 的外出连接计数机制结合时，数据包先被计数，然后再被传递给 snort_inline 分析。

下一步工作是配置snort_inline的规则集。Honeynet中使用snort_inline不是为了阻止所有的外出通信，仅是阻止外出的攻击，因此使用的规则集应该只含有已知的攻击规则，Honeynet Project提供了一个脚本snort_inline.conf[8]，用于把标准的Snort规则转换为snort-inline规则，可供参考。

2.3 数据捕获

下一步的工作是配置数据捕获机制。数据捕获的关键是在尽可能多的层次搜集信息，本例中的 Honeynet 定义了 3 个数据捕获层次：防火墙日志，网络通信和系统活动。

防火墙日志功能的实现很简单，这在数据捕获部分已经完成了，通过执行脚本rc.firewall[5,6]，就已经将所有的进入、外出Honeynet的通信记录在文件/var/log/messages中。这些日志信息最先表明攻击者的行为，在攻击者向外发起攻击时最先告警。

在网络通信层上，要记录每一个进入、离开Honeynet的数据包。虽然数据控制中snort_inline检查每个数据包，但作者不想在snort_inline中集中太多功能，因此，配置、运行另外一个进程来记录所有的数据包。在此配置一个标准的Snort[7]，使Snort捕获所有IP通信，并将其转储到日志文件tcpdump中以待后继分析。配置Snort时应将嗅探器与网关内部网口eth1 绑定，这样只会捕获进出Honeynet的通信；如果错误地将嗅探器与外部网口eth0 绑定，那么记录下的将不仅是与Honeynet相关的数据，还有外部网络的通信，这会“ 污染 ”捕获到的信息。

数据捕获的第 3 层次即捕获攻击者在 Honeypot 内的行为是最难实现的。多年以前，多数攻击者与目标系统相连时采用的都是明文协议，如 FTP、HTTP 或是 Telnet 等，因此可以在通信线路上监听攻击者的活动。但是，现在攻击者普遍采用了加密机制，使用 SSH 协议、3DES 通道等与目标系统通信，所以无法在通信线路上捕获数据，而只能在系统内部获取攻击者的活动信息。绝大多数的加密信息都会在端系统解密，对于 Honeynet 系统来说，就是在 Honeypot 内解密，因此，当数据在 Honeypot 内解密后，再捕获数据，就可以绕过加密机制。

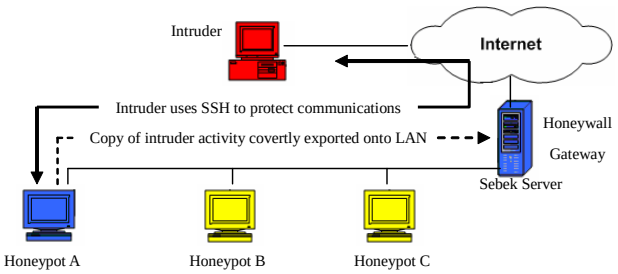

图 2 使用 Sebek 捕获攻击者活动

Sebek[9]实现记录系统活动的功能，它是Honeynet Project开发的内核态数据捕获工具。Sebek包含两个组件：客户端和服务器；客户端安装在Honeypot上，在系统内核运行，能够记录系统内所有用户的活动，然后把这些数据通过UDP协议传递给服务器；本例中，服务器安装在网关Honeywall上，它聚集客户端传来的数据；通过分析这些数据，就可以重构攻击者的行为；Sebek的安装结构如图 2 所示。Sebek安装完毕后，要进行配置，其配置文件指明要搜集哪些信息，如何将数据传到服务器。同一个Honeynet内的所有Honeypot的Sebek配置文件应该一致。

2.4 自动告警

Honeynet的最后一个需求是告警，即确保Honeynet被侵入时，管理者能得到及时的通知。Swatch[10]是实现自动告警的一个工具，把它安装在网关Honeywall上，它按照配置文件所描述的模式监视防火墙的日志文件/var/log/messages，当发现有符合某个模式的日志时就通过电子邮件、系统响铃等方式发出警告，或者运行预先设置好的命令或程序。只有很好地理解入侵，才能在规则中全面、正确地包含入侵特征，建立完整的报警规则集。每个从Swatch发出的报警包含数据包的源地址、目的地址、事件发生的时间及其它一些信息。

Honeynet 的主要功能设置完毕，接下来可以在 Honeypot 中放置一些攻击者感兴趣的资源、或者是你想要研究其安全性的资源，后面的工作就是等待攻击者。

3 Honeynet 的风险

为了获得攻击者的信息，必须允许攻击者访问Honeynet中资源，因此Honeynet存在一些特有的风险，主要有以下 4 方面：危害，探测，终止和妨碍[1~3]。

(1)危害是指攻击者侵入 Honeynet 以后，利用 Honeynet 的资源向外发起攻击，以危害其它的系统。

(2)探测是指攻击者可能识破 Honeynet 的身份，此后，把错误、伪造的信息引入 Honeynet，从而误导下一步的数据分析，使其使用价值显著降低。

(3)终止是使 Honeynet 丧失功能，攻击者通过破坏数据控制或捕获功能就可以做到这一点。

(4)最后一种风险是妨碍：攻击者进入 Honeynet 后，可能利用 Honeynet 系统进行非法活动，如攻击者利用 Honeypot 上载并散布一些违禁、非法的资料，如电影、音乐的非法拷贝、偷来的信用卡号等。这些行为并不是对其它 Honeynet 以外系统的攻击，但 Honeynet 的设置者将不得不证明事实上他不应该对这些行为负责。

多层次的数据控制和捕获机制可以降低上述风险，同时还应使用人工监视和定制系统。但这些方法只能降低 Honeynet 的风险，而不能彻底消除，因此作为一种安全技术，Honeynet 不能取代其它的技术，而应该配合使用。

参考文献

1 Spitzner L. Honeyptos: Definitions and Value of Honeyptos[Z]. 2003-05. www.tracking-hackers.com.

2 Honeynet Project. Know Your Enemy: Honeynets[Z]. 2003-11. www.honeynet.org.

3 Honeynet Project. Know Your Enemy: GenII Honeynet.[Z]. 2003-11. www.honeynet.org.

4 Honeynet Project. Know Your Enemy: Honeywall CDROM[Z]. 2003-05. www.honeynet.org..

5 杨沙州. Linux Netfilter 实现机制和扩展技术[Z]. 2003-10. www-900.ibm.com/developerWorks/cn/linux/.

6 Rob McMillen. Rc.firewall[Z]. 2002-04. www.honeynet.org/tools/dcontrol/.

7 Martin Roesch. Snort Users Manual[Z]. 2002-04. www.snort.ort.

8 Honeynet Project. Snort_inline.conf[Z]. 2003-12. www.honeynet.org/tools/.

9 Honeynet Project. Know Your Enemy: Sebek - a Kernel Based Data Capture Tool[Z]. 2003-11. www.honeynet.org.

10 Swatch, the Simple Watch[Z]. 2003-05. sourceforge.net/projects/swatch/.