摘 要

日益突出的信息安全问题已成为 Internet 众多研究中的一个热门课题。面对日益复杂的入侵事件，被动防御很难满足当前网络安全的需求。基于蜜网的主动防御的安全体系逐渐成为网络安全技术研究关注的焦点。主动防御技术是动态防御技术，它能够对网络的信息进行实时监控、捕获、分析，能够完成牵制和转移黑客的攻击，可以从中了解到敌人的动机和入侵方法、策略，还能对网络入侵进行分析取证甚至跟踪黑客。

在讨论蜜网系统的基础上，结合当前流行的入侵检测技术，设计并实现了一个基于 Honeynet 技术的主动安全防御系统。就蜜罐定义及蜜网系统体系结构，详细阐述了蜜网系统的三大核心技术：数据控制、数据捕获、数据分析。介绍基于内核层调用的 sebek 如何捕获入侵者加密信息。深入研究入侵检测系统 Snort 的运行体系结构模型及其日志数据库关系结构，并结合 Base 实现安全基本分析引擎审计系统；调查研究当前无线网络安全状况及介绍无线网络的认证加密算法，同时利用 Honeyd 部署一个虚拟无线蜜罐系统。通过深入分析部署蜜网系统数据发现跟踪僵尸网络，利用 Nepenthes 深入分剖析恶意软件攻击传播详细过程。

通过基于蜜网的主动防御安全系统的研究及功能测试，可以实现防御无线网络安全的入侵。实验表明，主动防御安全系统对分析僵尸网络的控制机制及发现未知恶意软件如何利用知名的漏洞进行传播具有实用价值。

关键词：主动防御　蜜网　蜜罐　入侵检测　僵尸网络

Abstract

Information security has become one of the many heated problems in the area of Internet research. Facing increasingly complex intrusion events, passive defense is unable to meet the challenges posed by the current network security issues. Active defense systems on the basis of honeypots are gradually becoming a focus in the filed of network security research. The active defense technique, as a dynamic defense measure, is able to monitor, capture, and analyze the online information in real time; to contain and transfer the attacks; to analyze the motivation, methods and tactics of the attackers; to collect the forensic evidence of the intruders and even track the hackers.

By analyzing the honeynet system, a new active information security defense system, combining the currently popular Intrusion Detection techniques, is designed and implemented. The key technique of the active defense system is the honeynet. In addition to discussing the concept of honeypot and presenting the systematic structure of the honeynet system, this thesis illustrates in detail the three critical techniques for the honeynet system: data capture, data control and data analysis; it also introduces how sebek, a kernel call, captures the encrypted information from the intruders; it then investigates in depth the structural model of Snort, an open source intrusion detection system, and its log database, used to implement an auditing system together with BASE (Basic Analysis and Security Engine). The current status of the wireless networks is surveyed, and the related algorithms for authentication and encryption are also introduced. Subsequently, a virtual wireless honeypot system with the use of Honeyd is designed and deployed.

The anti-intrusion protection for wireless networks is implemented with the research and functional test of the active defense systems based on honeynets. It is shows by the experiments that the active defense system is valuable in analyzing the mechanism of network zombies and discovering how unknown malicious software is distributed by exploiting the known vulnerabilities.

Key Words: Active Defense Honeynet Honeypot
Intrusion Detection Botnet

1 绪 论

1.1 研究背景及意义

随着网络日益广泛的使用，随之而来的信息安全问题日益严重。通过网络犯罪而对各个方面所造成的危害也日益严重，网络安全已经成为当今最为关心和棘手的问题。从大的方面来说，网络安全问题已经威胁到国家的政治、经济、军事、文化、意识形态等诸多领域。近十几年以来，网络上的各种安全性问题越来越多，也越来越严重。对 50 个国家的抽样调查结果显示：2000 年有 73%的单位受到各种各样形式的入侵，而在 1996 年，这个数字还是 42%[1]。现在世界上每年因利用计算机网络进行犯罪所造成的直接经济损失之大令人咂舌。据有关方面统计，目前美国每年由于网络安全问题而遭受的经济损失超过 170 亿美元，德国、英国也均在数十亿美元以上，法国为 100 亿法郎，日本、新加坡问题也很严重[2]。在国际刑法界列举的现代社会新型犯罪排行榜上，计算机犯罪也已名列榜首。

面对网络安全种种威胁，对于研究工作者来说，面临最大的问题及挑战之一就是缺乏对入侵者的了解，研究者需要了解攻击者，要了解存在的威胁并保护网络资源，我们需要了解敌人，了解是谁正在攻击，攻击的目的是什么，攻击者是如何进行攻击，攻击者使用什么方法及工具攻击，以及攻击者何时进行攻击等。要战胜敌人，就要做到知已知彼－了解你的敌人。要想有效地打击计算机犯罪，更好地保护资源，不仅仅要了解黑客的攻击技术和方法，还要深入了解黑客的攻击策略和动机。敌暗我明，攻击者与防御者之间在进行着一场非对称的博弈，特别是信息上的不对称，攻击者可以利用扫描、探测等一系列技术手段全面获取攻击目标的信息，而防御者对他所受到的安全威胁一无所知，即使在被攻陷后还很难了解攻击者的来源、攻击方法和攻击目标。因而在对抗中我们必须采取主动防御，由于攻击者是一个非常好斗的群体，设置陷阱很容易让他们自投罗网，然后再利用相应的技术监视并记录攻击者的所有活动。安全人员可以从黑客的具体行为中分析他们所使用的工具、方法和手段，研究他们的攻击策略、动机，了解他们入侵一个系统后做些什么及进行入侵追踪取证等。世界头号黑

客米特尼克就是中了安全专家下村努在服务器中设计的蜜罐陷阱而遭追踪被捕入狱[3]。

传统的网络安全技术对于网络攻击，广泛使用的安全防护手段如防火墙、入侵检测系统等被动防御手段，面对日益复杂和千变万化的攻击事件来说，这些被动防御技术逐渐变得力不从心。因此，一种新型的主动防御技术－蜜罐及蜜网技术，逐渐成为网络安全技术研究方面的热点。主动防御是指在动态过程中，直接对网络信息进行监控，能够完成牵制和转移黑客的攻击，对黑客入侵方法进行技术分析，对网络入侵进行取证甚至对入侵都进行跟踪。

本文在讨论蜜网系统的基础上，结合当前流行的入侵检测技术，研究并实现了一个基于 Honeynet 技术的主动安全防御系统。通过部置一些作为诱饵的主机、网络服务以及信息诱使攻击者对他们进行攻击，减少对实际系统所造成的安全威胁，更重要的是蜜罐及蜜网技术可以对攻击行为进行监控和分析，了解攻击者所使用的攻击工具和攻击方法，推测攻击者的意图和动机，从而能够让防御者清晰地了解他们所面对的安全威胁。本主动防御安全系统提供了高效收集数据的能力和极低的误报漏报率，虽然收集的资料很少，但资料都具有很高的价值。本文研究的实时安全基本分析引擎审计系统，本系统是分布式协同入侵检测的控制中心，基于友好中文管理界面，实现实时入侵警告，分类查询生成各种中文图形报表。构建无线蜜罐主动安全防御系统可以及早发现无线网络中的安全死角，保护无线网络不受恶意的入侵。实验测试表明，主动防御安全系统对分析僵尸网络的控制机制及发现未知恶意软件如何利用知名的漏洞进行传播具有实用价值。

网络安全遵循“木桶原理”，即一个木桶的容积决定于组成它的最短的一块木板，一个系统的安全强度等于它最薄弱环节的安全强度。通过蜜网构建主动防御安全体系能形成立体的防御堡垒，解决网络安全中最短的那根木条。

1.2 研究现状

蜜罐和蜜网是实现主动防御安全系系统的关键技术。蜜罐的思想最早出现在九十年代初期，由网络管理员所应用，通过欺骗黑客达到追踪的目的。1998 年著名计算

机安全专家 Fred Cohen 发布了著名的蜜罐工具 DTK（欺骗工具包），并于 2001 年在理论层次上给出了信息对抗领域欺骗技术的框架和模型，Fred Cohen 的研究工作也为蜜罐技术奠定了理论基础[4]。

蜜罐技术(Honeypot)是一个故意设计为有缺陷的系统，一种欺骗入侵者以达到采集黑客攻击方法和保护真实主机目标的诱骗技术。

蜜网技术(Honeynet)则是由蜜罐技术逐步发展过来的，但与传统蜜罐技术的差异在于，蜜网构成了一个黑客诱捕网络体系架构，在这个架构中，我们可以包含一个或多个蜜罐，同时保证了网络的高度可控性，以及提供多种工具以方便对攻击信息的采集和分析[5]。

目前在蜜网技术上最为活跃的是“蜜网项目组”，“蜜网项目组”是一个非赢利性的研究组织，其目标为学习黑客社团所使用的工具、战术和动机，并将这些学习到的信息共享给安全防护人员。

此外，“蜜网项目组”还倡议成立了“蜜网研究联盟” （Honeynet Research Alliance），以促进蜜网技术的发展及成果共享。

国内北京大学计算机研究所信息安全研究中心的蜜网研究项目“狩猎女神”项目也于 2005 年 2 月成功加入世界“蜜网研究联盟”，成为国内第一个加入蜜网研究联盟的研究团队。

1.3 本文的内容结构及主要研究内容

本文第一章绪论，第二章详细论述基于蜜网的主动防御安全系统结构，第三章深入对主动防御安全系统结构模型研究，第四章是一个主动主动防御安全系统实现及测试分析，第五章工作总结。对现在蜜网系统理论进行系统的分析研究，部署一个智能型主动入侵防御系统，完成相应系统的设计及功能测试。

本文主要研究的内容：

1）蜜罐体系结构研究。

2）研究最新的蜜罐与蜜网技术发展，实际部署和维护蜜网，并对蜜网捕获的黑客攻击及垃圾邮件、间谍软件进行深入分析，增进对蜜罐与蜜网技术的理解，了解互

联网最新的安全威胁。针对蜜网的三大核心功能之一的数据分析，研究蜜网攻击数据统计分析及关联分析技术。

3）深入剖析入侵检测 Snort 的原理，并结合 Base 实现分布式安全基本分析引擎审计系统。

4）组建基于无线蜜网的主动防御安全系统及利用蜜网进行取证分析。

5）蜜网系统的功能测试及性能测试。

2 主动防御安全系统结构

本章介绍主动防御安全系统结构，蜜罐及蜜网系统是主动防御安全系统关键技术。主要是蜜罐的定义、分类及对比，蜜网的核心技术体系以及部署蜜罐及蜜网所带来的风险。

2.1 蜜罐系统

2.1.1 蜜罐系统定义

蜜罐(Honeypot)这个词最早出现在1990年著名黑客传记小说《杜鹃蛋》中，书中描述主人公利用蜜罐诱骗追踪黑客[6]。蜜罐是一个包含漏洞的系统，通过模拟一个或多个易攻击的主机，给黑客提供一个容易攻击的目标。蜜罐另一个用途是拖延攻击者对真正目标的攻击，让攻击者在蜜罐上浪费时间，最初的攻击目标受到保护，真正有价值的内容将不受侵犯。蜜罐的最初目的之一是为起诉恶意黑客搜集证据，这就是“诱捕”系统。

“蜜网项目组”（ The Honeynet Project ）的创始人 Lance Spitzner 给出了对蜜罐的权威定义：蜜罐是一种安全资源，其价值在于被扫描、攻击或攻陷，这就意味着蜜罐是用来被探测、被攻击甚至最后被攻陷的，蜜罐不会修补任何东西，这样就为使用者提供额外、有价值的信息。蜜罐不会直接提高计算机网络安全，相反还存在一定的安全风险，但它却是其它安全策略所不可代替的一种主动防御技术[7]。

2.1.2 蜜罐技术的发展历程

（1）蜜罐技术的发展历程可以分为以下三个阶段。

从九十年代初蜜罐概念的提出直到 1998 年左右，“蜜罐”还仅仅限于一种思想，通常由网络管理人员应用，通过欺骗黑客达到追踪的目的。这一阶段的蜜罐 实质上是一些真正被黑客所攻击的主机和系统。

从 1998 年开始，蜜罐技术开始吸引了一些安全研究人员的注意，并开发出 一些

专门用于欺骗黑客的开源工具，如 Fred Cohen 所开发的 DTK（欺骗工具包）、 Niels Provos 开发的 Honeyd 等，同时也出现了像 KFSensor、Specter 等一些商业蜜罐产品。这一阶段的蜜罐可以称为是虚拟蜜罐，即开发的这些蜜罐工具能够模拟成虚拟的操作系统和网络服务，并对黑客的攻击行为做出回应，从而欺骗黑客。

虚拟蜜罐工具的出现也使得部署蜜罐也变得比较方便。但是由于虚拟蜜罐工具存在着交互程度低，较容易被黑客识别等问题，从 2000 年之后，安全研究人员更倾向于使用真实的主机、操作系统和应用程序搭建蜜罐，但与之前不同的是，融入了更强大的数据捕获、数据分析和数据控制的工具，并且将蜜罐纳入到一个完整的蜜网体系中，使得研究人员能够更方便地追踪侵入到蜜网中的黑客并对他们的攻击行为进行分析[8]。

蜜罐可以按照其部署目的分为产品型蜜罐和研究型蜜罐两类：

产品型蜜罐的目的在于为一个组织的网络提供安全保护，包括检测攻击、防止攻击造成破坏及帮助管理员对攻击做出及时正确的响应等功能。一般产品型蜜罐较容易部署，而且不需要管理员投入大量的工作。较具代表性的产品型蜜罐包括 DTK、honeyd 等开源工具和 KFSensor、ManTraq 等一系列的商业产品。

研究型蜜罐则是专门用于对黑客攻击的捕获和分析，通过部署研究型蜜罐，对黑客攻击进行追踪和分析，能够捕获黑客的键击记录，了解到黑客所使用的攻击工具及攻击方法，甚至能够监听到黑客之间的交谈，从而掌握他们的心理状态等信息。研究型蜜罐需要研究人员投入大量的时间和精力进行攻击监视和分析工作，具有代表性的工具是“蜜网研究联盟”所推出的第二代蜜网技术[9]。

蜜罐还可以按照其交互度的等级划分为低交互蜜罐和高交互蜜罐：交互度反应了黑客在蜜罐上进行攻击活动的自由度。低交互蜜罐一般仅仅模拟操作系统和网络服务，较容易部署且风险较小，但黑客在低交互蜜罐中能够进行的攻击活动较为有限，因此通过低交互蜜罐能够收集的信息也比较有限，同时由于低交互蜜罐通常是模拟的虚拟蜜罐，或多或少存在着一些容易被黑客所识别的指纹（Fingerprinting）信息。产品型蜜罐一般属于低交互蜜罐。高交互蜜罐则完全提供真实的操作系统和网络服务，没有任何的模拟，从黑客角度上看，高交互蜜罐完全是其垂涎已久的“活靶子”，因

此在高交互蜜罐中，我们能够获得许多黑客攻击的信息。高交互蜜罐在提升黑客活动自由度的同时，自然地加大了部署和维护的复杂度及风险。研究型蜜罐一般都属于高交互蜜罐，也有部分蜜罐产品，如赛铁门公司的 ManTrap，属于高交互蜜罐。

（2） 蜜罐系统的对比

对于真实或是虚拟 honeypot 的选择方面，我们需要考虑的是风险和回报。虚拟 honeypot 比较廉价，但也有一定安全风险，它在抓住黑客方面做得没有真实的 honeypot 好。另一方面，虽然真实的 honeypot 在入侵检测方面比虚拟 honeypot 好很多，但最顶级的黑客有可能利用真实的 honeypot 入侵控制你的网络。

- 虚拟 honeypot 的优势

虚拟 honeypot 是一个仿真程序。比如虚拟 honeypot 一般可以仿真 FTP 服务器，并监视所有的 TCP 和 UDP 端口并记录所有端口的活动情况。当黑客发现这个虚假的（他本人不知道）FTP 时，就会试图开启一个 FTP 对话。这时虚拟 FTP 服务器（虚拟 honeypot）就会记录下这个黑客的所有活动。比如 honeypot 会记录下哪个端口被使用、采用何种认证机制等。而虚拟 FTP 服务器会和真正的 FTP 服务器一样对黑客的行为作出响应。更好的是，由于这是个虚拟的 FTP 服务器，它没有真正的操作系统，因此就算黑客攻入了 FTP，也不会进一步控制你网络中的其它电脑。理论上说，这个方法相当好，它使用起来相当安全，并且可以捕获大量的有用信息。比如，如果获取了黑客登录时的凭证，你就可以查出到底是哪个帐户被攻击了，这样就可以作出相应的补救动作。这些就是它的全部优势。

- 虚拟 honeypot 的劣势

对于虚拟 honeypot 来说，有两点最主要的不足。首先，它只能愚弄那些初级黑客。你要记住，虚拟 honeypot 并没有一个真正的操作系统支撑（有的解决方案中内嵌了简单的 Windows 或 Linux）。因此有经验的黑客会发现很多命令在这台主机中不起作用。这会使他立即知道自己进入的只是一台 honeypot，而不是真正的服务器。

虚拟 honeypot 的另一个不足是它记录的信息种类有限。比如一个虚拟 honeypot 伪装成 FTP 服务器，那么它就只能获取和 FTP 相关的信息。当然，大部分虚拟 honeypot 还可以获取端口扫描和其它一些基本的攻击信息。然而，如果一个黑客利用 IPv6 端

口发送加密的信息又会如何呢？由于虚拟 honeypot 功能有限，它无法记录这类的问题。简单说，虚拟 honeypot 可以检测并记录已知的攻击种类，但对于新型的攻击却没什么用处。

- 真实 honeypot 的优势

一个真正的 honeypot，是一个或多个真实的系统组成的诱饵系统。由于它是带有操作系统的真实系统，因此它对于黑客的操作响应与网络上其它主机完全一样。这有好处也有坏处。好处是，黑客几乎不可能察觉到他们已经进入了一个陷阱，而不是真正的实用网络。实际上，唯一能让黑客起疑心的现象就是那些不太完善的 honeypot 网络没有采取任何正常的安全更新措施。

真实的 honeypot 最大的优点就在于入侵检测能力。系统会假定任何发送到 honeypot 网络的数据都是带有恶意的，因此完全不用担心黑客会采用什么新方法而不被 honeypot 捕获。黑客的任何操作都会被真实的 honeypot 记录下来。

- 真实 honeypot 的劣势

真实的 honeypot 也有不足，它有可能被高级黑客征服而变为进攻你的正常网络的跳板。为了防止这种情况，你需要在 honeypot 网络与正常网络间架设防火墙，以阻挡二者间的任何数据通信。更复杂的 Linux honeypot 带有防止黑客入侵正常网络的功能，而对于 Windows 上的 honeypot，目前还没有类似的功能[10]。

2.2 蜜网系统

蜜网技术是由蜜网项目组(The Honeynet Project)提出并倡导的一种对攻击行为进行捕获和分析的新技术。蜜网技术实质上仍是一种蜜罐技术，但它与传统的蜜罐技术相比具有两大优势：首先，蜜网是一种高交互型的用来获取广泛的安全威胁信息的蜜罐，高交互意味着蜜网是用真实的系统，应用程序以及服务来与攻击者进行交互，而与之相对的是传统的低交互型蜜罐，例如 DTK 和 Honeyd，它们仅提供了模拟的网络服务。其次，蜜网是由多个蜜罐以及防火墙、入侵防御系统、系统行为记录、自动报警、辅助分析等一系列系统和工具所组成的一整套体系结构，这种体系结构创建了一个高度可控的网络，使得安全研究人员可以控制和监视其中的所有攻击活动，从而去

了解攻击者的攻击工具、方法和动机[11]。

Honeynet 是一个网络系统，而并非某台单一主机，这一网络系统是隐藏在防火墙后面，所有进出的数据都受到监控、捕获及控制。我们可以使用各种不同的作业系统及设备，如 Solaris, Linux, Windows NT, Cisco Switch 等等。这样建立的网络环境看上去会更加真实可信，同时我们还有不同的系统平台上面运行着不同的服务，比如 Linux 的 DNS server、Windows NT 的 webserver 或者一个 Solaris 的 FTP server、构建一个无线 AP 等。我们可以学习不同的工具以及不同的策略—或许某些入侵者仅仅把目标定於几个特定的系统漏洞上，而我们这种多样化的系统，就可能更多了揭示出他们的一些特性。

- 蜜网的价值

从传统意义上来说，信息安全一直都是防御性的。防火墙、入侵检测系统、加密，所有这些机制都是用来对自己的资源进行防御性保护的。这里的策略就是尽可能好地保护自己的组织，在防御中检测出失误，然后对失误进行回击。这种方法的问题在于，这纯属被动的防御，而敌方处于进攻地位。Honeynet 试图扭转这种局面，赋予组织以主动权。其主要目的在于搜集敌方的情报。这样，组织就有可能将一场攻击或者防御中的失误扼杀在萌芽状态。人们常常会把信息安全和军事进行比较，安全研究者对敌人了解得更多，就更有主动权。

2.2.1 蜜网的体系结构

蜜网是由多个蜜罐组成的，图 2-1 给出目前我们部署的蜜网网络拓扑结构图。

在蜜网中，部署了分别安装 Red Hat Linux Fedora 3、Windows2003、Windows XP 系统 3 台物理蜜罐，另外通过 Vmware、Honeyd 等虚拟蜜罐工具模拟了多台虚拟蜜罐主机。

蜜网网关（Honeywall）是基于蜜网项目组最新的第三代蜜网技术－Roo CDROM 构建，以桥接方式对全部的网络流量进行捕获和控制，并提供 Walleye 数据分析界面，HoneyWall 的管理接口连接一台控制机，供操作员对蜜网进行维护和攻击案例分析。

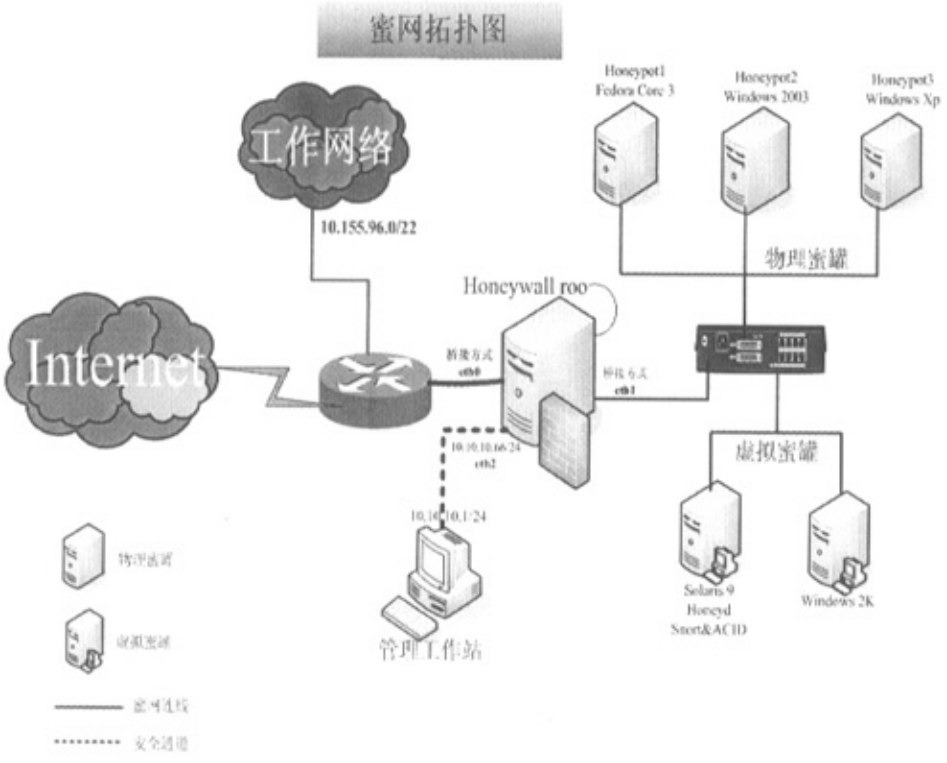


图 2-1 蜜网拓扑图

任何 Honeynet 重要的因素是网关—用于把 Honeynet 从外部世界隔离开来。网关作用类似墙，实际上我们在 Honeynets 中把网关称为 Honeywall。所有进出 Honeynet 的通信必须通过这个 Honeywall。这其实就变成你的 Honeynet 命令和控制中心，所有神奇的东西都才这里发生。从图 2-1 中看到部署的 Honeynet 体系架构，在这个例子中，部署的网关是一个两层的网桥，当然也可以使用一个三层的路由网关，不过最好还是使用网桥，因此这会使这个网关比较难检测出来。在拓扑图中，部署的 Honeynet 在内部 10.155.96.0/22 网络中进行配置。在以前，多数 Honeynets 一般传统的配置在外网或边界网络上。通过使用第二层的网关，Honeynets 现在就可以象我们这样集成在内部网络中。这允许我们能跟踪和了解外部网络的威胁，也可以知晓潜在的内部安全问题。

Honeywall 把产品系统从引诱目标的 Honeynet 网络中隔离开来。我们网关(eth0)的外部接口连接真正工作系统网络，eth1 网关的外部接口连接 Honeynet 系统网络，由于这是个网桥，因此外部和内部系统在同一个 IP 网络中。我们也可以拥有第三个接口(eth2)，这个接口我们可以用于远程管理网关，包括移动任何日志或在集中点捕获任何数据。外部和内部接口由于在网桥模式下，因此它们不需要指定任何 IP 地址。但是第三个接口(eth2)必须有一个指定的 IP 栈，如指定一个 IP 地址为 10.10.10.66，不过这个网络必须独立的，安全的网络，以便安全的进行管理。这个结构的好处是网关由于没有路由跳，没有 TTL 递减，没有任何 MAC 地址关联网关而比较难被探测，所有通过的数据都是透明的。我们也可以在单一网关上通过结合数据控制和数据捕获简单的进行 Honeynet 配置。下一步是我们来构建网关来支持这个结构。我们使用最小化，安全加固的 Linux 系统作为网关，这个 Linux 系统必须是一个可信任系统是尤为重要的，必须保证攻击者不能访问我们的网关。

2.2.2 部署蜜网的三大核心技术：数据控制、数据捕获和数据分析

1）数据控制

数据控制的目的是防止攻击者使用 Honeynet 攻击或破坏非 Honeynet 系统。数据控制可减低威胁程度，但不能完全消除此带来的威胁。对于数据控制，你必须回答一个问题是你必须控制多少外出的活动通信？你允许攻击者操作的行为越多，你就能了解的更多。但是，这也会造成潜在的威胁和破坏会更多。因此你必须允许攻击者的操作不能伤及其他人，但又不能使自己学到的内容很少。允许攻击者能拥有多少的活动范围依据你能承受多少的威胁程度。为了使 Honeynet 更具挑战性，我们必须使攻击者不知情的情况下牵制攻击者。要完成这个任务，我们必须实现两个技术，连接计数和 NIPS。连接计数是 Honeynet 初始化时我们限制多少外出的连接，NIPS(网络入侵防止系统)可以阻挡(或关闭)已知攻击。组合这两种技术可建立一个冗余的可伸展性的数据控制机制， 我们会在第二层网关实现这两种技术，我们在网关上实现数据控制是由于这地方是所有外出和进入数据通信的总通道，是攻击者操作行为的要点。图 2-2 图是蜜网网关 HoneyWall 数据控制机理图[12]。

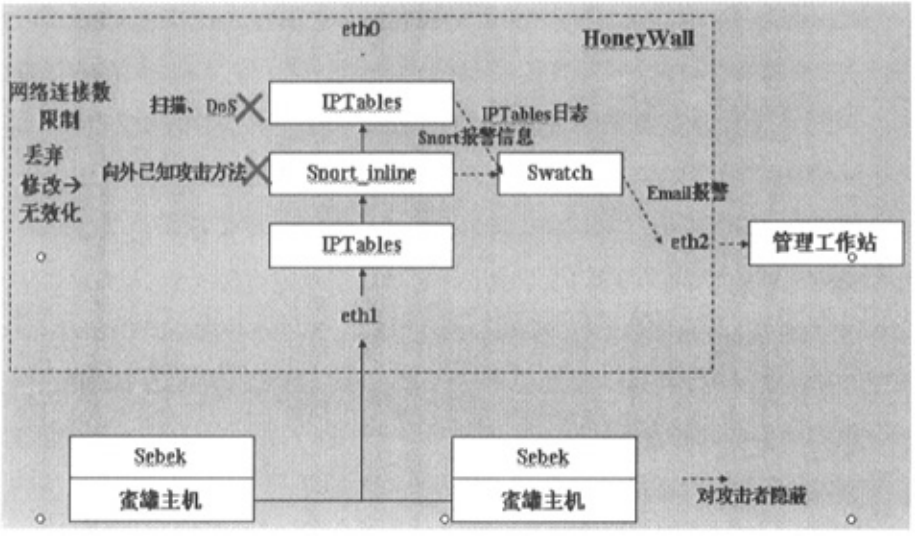


图 2-2 数据控制机理图

在数据控制中我们主要使用 Iptable 实现连接限制，我们限制攻击者可以从 honeypot 发起的外出连接，这里的目的是对外出连接进行计数，当部分计数限制连接被检查到后，就会阻挡之后更多的通信。这最主要的是能降低攻击者利用 honeynet 进行过多的扫描滥用或进行拒绝服务攻击等许多需要外出连接的通信。我们通过 rc.firewall 脚本进行配置和实现，使用 IPTable 来进行限制，在这个脚本中，可以设置攻击者可以初始化多少 TCP，UDP，ICMP，或者其他外出连接。当然允许多少外出连接取决于你能接受的威胁程度。 限制从 honeynet 外出连接的限制可以防止攻击者从 Honeynet 进行扫描或对其他系统进行大范围的攻击，或者进行拒绝服务攻击，当对外出初始化进行连接后会很大程度上降低破坏的程度。但是，必须清醒认识到的是这也可以成为一个 honeynet 的特征，一个攻击者可以通过初始化多个外出连接连判断是否是一个 Honeynet。rc.firewall 脚本中的连接限制如下所示，注意变量 OTHER 是任意 IP 协议，而不是 TCP，UDP 或者 ICMP(如 IPsec, IPv6 tunneling, Network Voice Protocl 等)。

除了使用 IPtable 的 rc.firewall 脚本实现连接限制，同时还可以通过 Snort_inline 实现NIPS,NIPS的目的是识别和阻挡已知的攻击，它通过查看每个通过网关的包实现。

如果某个包匹配某个 IDS 规则，不仅会生成警告（如传统的 IDS），也可以是包丢弃（阻挡攻击）或者进行修改（取消攻击行为）。个 NIPS 好处是我们可以动态的减低外出攻击行为的成功率，此 NIPS 不好的一点就是只能检测已知的攻击。

为了配置好目标，我们需要安装丢弃规则库和 /etc/snort_inline 中的 snort_inline.conf 配置文件，我们可使用与/etc/snort 独立的目录以更好的区分。要运行 snort_inline，我们必须使用 snort_inline 启动脚本，这个脚本预配置了所有变量用于 snort_inline 启动和运行。

2）数据捕获

当我们配置完数据控制后，就可以配置数据捕获了，数据捕获的目的是记录攻击者所有的活动，这是 Honeynet 的最终目的，收集信息，没有数据捕获，Honeynet 将没有任何价值。数据捕获的关键是在尽可能多层收集信息。没有靠一层能获得所有信息。如需要人认为我只要记录攻击者的所有击键记录就可以了，但是这往往不是想象的这么简单，当攻击者利用工具进行操作时，如果不对工具的通信进行捕获，你怎么知道此工具具体在干什么操作呢？Honeynet 项目组认为如下三个方面是数据捕获重要的来源：防火墙日志，网络通信及系统活动。以下将描述怎样对这三方面进行实现：

防火墙日志非常简单，我们已经完成这一步，通过 rc.firewall 脚本的实现，我们已经可以在/var/log/messages 中记录所有的进入和外出的连接。这些信息非常重要，因为这是我们第一个指示信息能判断攻击者到底要做什么，当攻击者发起外出连接时也是第一个警告的地方。基于以有的经验，防火墙日志能很快的识别新或未知行为，脚本能识别四种不同的通信：TCP, UDP, ICMP 和 OTHER，就象数据控制中，OTHER 代表任意非 IPproto 1, 6, 或 17 通信，当某些人使用非标准 IP 通信时，很可能是攻击者在尝试新的攻击或没有公开的方法[13]。

第二个元素是捕获每个出入 Honeynet 的包及包的负载，snort_inline 进程当然可以完全处理此任务，但是我们不想把所有鸡蛋放到一个篮子里，相反的是我们配置和运行另外一个进程来捕获所有这种活动，我们通过使用标准 snort.conf 配置文件来完成。这个配置文件捕获所有 IP 通信并生成 tcpdump 日志文件以便以后的进一步分析。

我们也需要每天交替捕获的日志，这个可以通过 snort.sh 启动脚本来完成，注意这里重要的是我们把启动脚本绑定在内部接 eth1 上，如果错误的把它绑定在外部接口上，不仅会记录 Honeynet 数据，也会记录所有相关外部网络的其他通信。这一定程度上会混乱你捕获的数据。通过内部接口进行捕获，你就会仅仅获得 Honeynet 上的你所需要的通信。另外一个启动脚本有利之处是标准化记录数据的位置，如果你有多个 Honeynets 进行日志记录，这将会变的比较重要。

第三方面是最具挑战性，在蜜罐中攻击者的所有活动，几年前这工作非常简单，因为远程交互全是通过明文协议如 FTP，HTTP 或 TELNET，研究者只要嗅探连接的击键记录，但是攻击者采用了加密手段，现今攻击者使用 SSH 或 3DES 通道对入侵的计算机进行通信，研究者就不能再直接从线上记录击键，而需要从系统自身获得。此系统有利的一点是多数加密是在系统末端进行解密，如在我们的蜜罐上进行解密。如果我们从蜜罐上捕获这些解密的数据，我们就可以绕过加密通信。Sebek 就是设计用于此目的的工具，Sebek 是一个可隐藏内核模块可记录攻击者的活动，一旦安装在蜜罐上，Sebek 客户端运行在内核上，通过 Sebek 收集的信息不存在在攻击者容易发现的蜜罐上，因为 Sebek 客户端会通过 UDP 传送数据给窃听的机器(如 Honeywall 网关或远程另外网络中的系统)[14]。因此 Sebek 在蜜罐中是隐藏的，因此攻击者不能知晓是否这些包被窃听。而且如果攻击者下载或使用他们自身的窃听工具，Sebek 必须不能让这些窃听工具给记录，这可以通过修改蜜罐使它不能嗅探到任意预指定的端口号通信信息。Sebek 然后简单在通信上集取信息。由于所有蜜罐由 Sebek 控制，没有其他人可以窃取获得线上通信的击键。不过如果你的蜜罐没有安装 Sebek，或者 Sebek 不正确配置，攻击者控制了此系统后，他们可以窃听由于没有隐藏并来自其他系统的 Sebek 包信息。要 Sebek 运行在内核中，它必须以特定 OS 和内核版本进行编译。

1）Sebek 结构

Sebek 有两个组成部分：客户端和服务端。客户端从蜜罐捕获数据并且输出到网络让服务端收集。图 2-3 是 Sebek 的系统结构图。

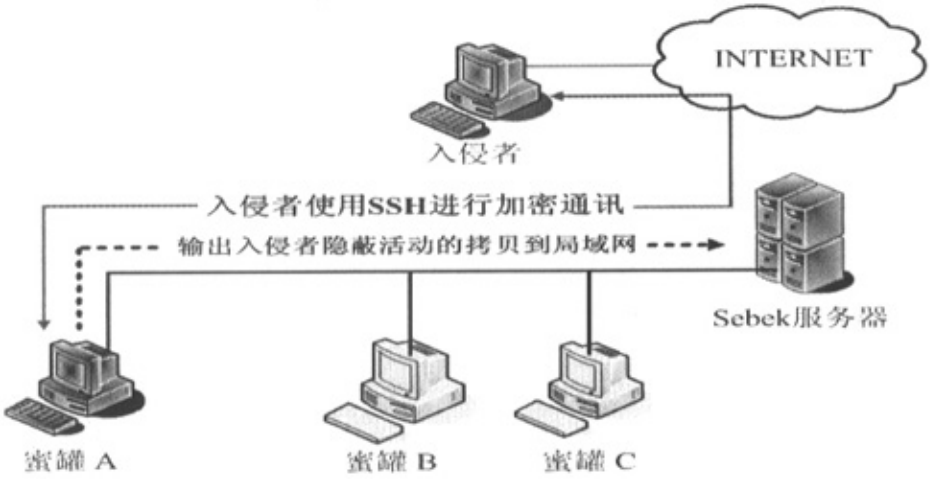


图 2-3 Sebek 体系结构图

2）客户端数据捕获

数据捕获是由内核模块来完成的，我们使用这个模块获得蜜罐内核空间的访问，从而捕获所有 read()的数据。Sebek 替换系统调用表的 read()函数来实现这个功能，这个替换的新函数只是简单的调用老 read()函数，并且把内容拷贝到一个数据包缓存，然后加上一个头，再把这个数据包发送到服务端。替换原来的函数就是改变系统调用表的函数指针[15]。

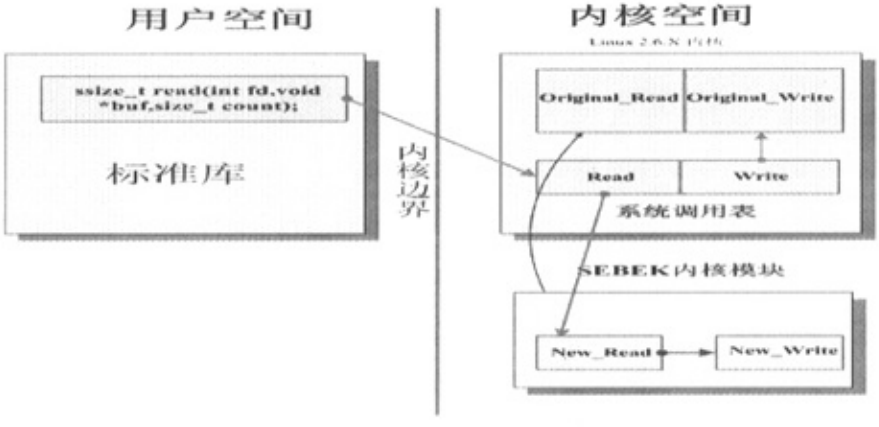


图 2-4 改变 read 系统调用的示意图

当用户空间的进程调用标准的 read()函数的时候会产生一个系统调用，这个调用映射到系统调用表数组的索引偏移。因为 Sebek 把 read 索引的函数指针修改指向到它自己的函数实现，所以当执行到内核的 read 实现时就会执行 Sebek 修改后的 read 调用，Sebek 通过这个系统调用看到所有访问的数据。我们可以用同样的技术来修改我们想要监视的任意系统调用。

数据在使用的时候一般都是解密的。比如有动作发生的时候，SSH 会话的击键显示在 shell 上的数据肯定不是加密的。这个动作一般会产生一个系统调用，在内核空间收集数据，我们能截获这个系统调用访问进程解密后但还没有处理的数据，通过这种方法就可以在加密环境捕获击键、文件传输、Burneye 的口令等。

3）客户端模块隐藏

为了使 Sebek 模块更加隐蔽，我们使用了一些如 Adore 等基于 LKM 的 rootkit 的技术。由于 Sebek 完全驻留在内核空间中，那么我们可以通过安装第二个 cleaner 模块来隐藏 Sebek 模块，这个模块把 Sebek 从安装模块链表中删除记录。移除模块有两方面的效果。首先，用户将看不见 Sebek 被安装了。第二，如果已经安装，用户将无法卸载 Sebek 模块。安装脚本 sbk_install.sh 的"Testing"变量如果设置成"1"，那么就禁止了隐藏功能。

4）客户端数据包输出

当 Sebek 客户端捕获数据，那么它需要在入侵者没有察觉的情况下把数据发送到服务端。一般情况下蜜罐都是布置了一个局域网内，如果 Sebek 只是简单使用 UDP 流来给服务端发送数据，入侵者只需监听网络上的数据传输就可以判断是否安装了 Sebek。不过 Sebek 还是使用 UDP 来给服务端发送数据，但是它修改内核使用户无法看到这些数据包，包括其它主机发送的该类型使用相同配置的数据包。而且当 Sebek 把这些数据发送到网络的时候，系统也无法阻止这些数据包的传输。

如果一个局域网上每个蜜罐安装了 Sebek，它们都不能发现任何 Sebek 数据，然而服务端能够完全访问这些由客户端捕获的数据。每个 read()调用请求 Sebek 都会产生一个或多个日志数据包，每个数据包都包含了一点关于这个调用内容的信息和这个调用访问的数据。每个包还包含了一个 Sebek 记录，这个记录包含一些产生调用的进

程描述、调用产生的时间和记录数据的大小。这些报完全由 Sebek 产生，而不是使用 TCP/IP 协议栈来产生或发送数据包，所以系统无法看到或阻断这些数据包。当数据包创建好的时候就直接发送给驱动设备，这就绕过了原始套接字代码和包过滤代码。由于嗅探器是基于 libpcap 的，而 libpcap 使用原始套接字接口来收集数据包，所以嗅探器不能看到运行 Sebek 主机上由 Sebek 产生的数据包。如图 2-5（注意 Sebek 产生的数据包怎样绕过协议栈并且直接发送到网络设备驱动。这使得攻击者检测 Sebek 的活动非常困难。）

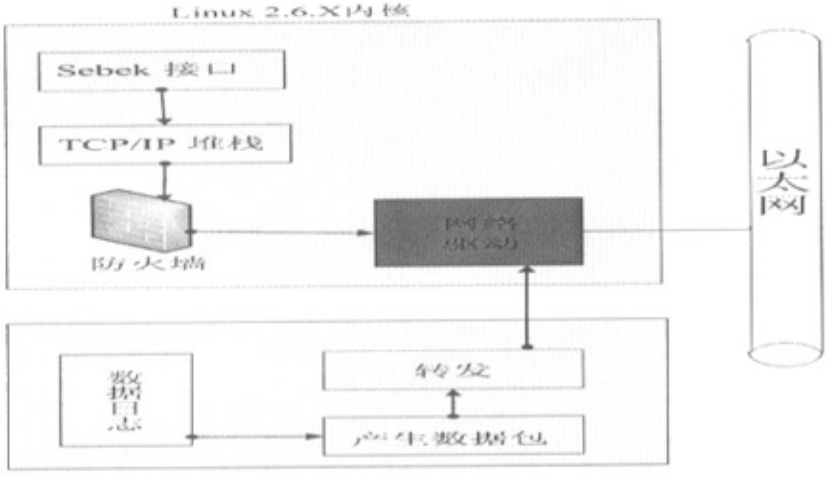


图 2-5 Sebek 数据包产生的概念图

因为 Sebek 产生自己的数据包并且直接通过设备驱动发送它们，所以用户无法使用 IPTABLES 来阻断数据包，也无法用网络嗅探器来监视它们，这也防止了蜜罐上的入侵者通过检查局域网数据传输来检测是否有 Sebek。第二个要解决的问题是需要阻止蜜罐 A 检测到蜜罐 B 的 Sebek 数据包。使用以太交换不能解决这个问题。因为 Sebek 不是通过 ARP 获取目标 IP 地址对应的目标 MAC 地址，所以它对 ARP 欺骗有自然免疫能力。但是在有些情况下，A 可以看到 B 的数据包，这样入侵者就可以在蜜罐 A 运行嗅探器来看到局域网上来自蜜罐 B 的 Sebek 数据包。

为了解决这个问题，Sebek 安装了自己的原始套接字实现。在 Sebek 数据包头定

义了预先设定的目标 UDP 端口和固有的魔术数字，如果这两个值都匹配了，那么这个数据包就会被忽略。这个简单的实现不会对 Sebek 数据包有影响，只是丢弃它们并且移到队列里的下一个数据包，这使得入侵者即使使用嗅探器也无法捕获 Sebek 数据包。

3）数据分析

蜜网的研究手段主要是通过数据捕获到的数据进行分析，这是最重要也是最关键的一步，通过网络日志了解哪些人何时何地进行入我们的蜜罐系统，通过分析 snort_inline 日志所有进出的连接以所有入侵的警告。通过分析以便发现新的 IDS 规则。通过分析蜜罐的系统日志及状态，查看 Sebek 记录，这些通过由 Sebek 内核模块捕获的击键记录。这些包需要被网络嗅探器(Snort)捕获然后能从我们刚才分析的相同的两进制日志文件中恢复出来，发蠕虫病毒和攻击者怎样入侵系统，通过数据分析，过滤出通过网关系统的所有垃圾电子邮件，收集垃圾邮件样本并分析特征。

通过分析无线蜜网系统，找出无线 AP 存在的漏洞和风险。使用 Snort 结合 Base 进行数据的统计分析，实时监控整个网络的安全状况。利用 mwcollect 和 nepenthes 实现对恶意软件及蠕虫的发现。

使用 Tcpdump,Sniffer, Ethereal 等数据抓包工具，结合 libcap 进行编程实现数据数据捕获及分析。数据分析将在后面章节中再详细介绍。

2.3 安全风险

2.3.1 蜜罐及蜜网的安全风险

蜜网是一个强有力的工具。他们能够收集到详细的有关各种安全威胁的信息。为了获得这些信息，安全研究者必须允许攻击者访问你的系统和引用程序，而且有可能是有特权的访问，为此付出的风险代价就非常高。任何由人开发的技术都可以被人打败。风险对不同的组织来说意义不同，组织必须清楚哪些风险是最重要。同时，不同组织对风险的承受能力也不同。安全研究者无法告诉组织错对是非，组织必须自己做出决定，安全研究者能做的只有让组织注意到这些风险。安全研究者将涉及以下四种

主要的安全风险：危害（ harm ），侦测（ detection ），失效（ disabling ），滥用（ violation ）。

- 危害是指当一个蜜网被用来攻击或伤害其他的非蜜网系统是的危害。
- 其次，你要冒着被侦测的风险。一旦一个蜜网的真实身份被确认，它的价值就将急剧降低。
- 第三，你还要冒着蜜网某些功能失效的危险。这可能是针对数据控制或者是数据捕获功能的攻击。攻击者也许不仅仅是想要识别出一个蜜网的身份，更想要使它的数据控制或捕获能力失效，可能还想让蜜网的管理员都不知道它失去了某些功能。
- 第四个也是最后一个风险是滥用，它是对其它的所有存在的风险的总称。攻击者可能从被攻陷的你的蜜网上尝试一些不用实际的攻击外部系统的犯罪行为。

2.3.2 蜜罐及蜜网的安全风险解决方法

在上面的四种情况中，有两个步骤可以帮你减轻风险：人工监控和定制。对于人工监控，我们指的是雇用一个受过训练的专业人士来实时监控和分析你的蜜网，任何时候当你怀疑个攻击者已经成功地获得（或者尝试获得）了你的某个蜜罐的访问权限时（比如检测到连出的连接，频繁建立的连入的连接，连入流量的增加，文件的传输，不正常的系统活动等等），一个安全专家应该监控和分析所有捕获的数据。其次，定制是很关键的。所有蜜网技术都是开源的，这意味着任何人都可以获得这些信息，其中包括那些我们认为正在积极地阅读它并开发对策的黑客界。蜜网技术有助于减少风险，但是最好的工具就是人，让人来参与监控，分析和响应蜜网中的活动。我们无论采取了何种措施，风险都依然存在。尽管我们已经竭尽全力去降低这种风险了，但是永远都不能完全消除。正如军事上常说的那句话“永远都不要轻敌”[16]。

2.4 本章小结

本章主要对蜜罐、蜜网的定义。蜜罐系统的分类对比，蜜罐与蜜网的关系作的阐述。重点介绍了部署蜜网系统的三大核心技术：数据控制、数据捕获和数据分析。蜜网是一种高交互的蜜罐。它最基本的优点是它能收集广泛的关于安全威胁的信息。一

个蜜网是一个类似于玻璃鱼缸的透明体系结构，在这个结构内部你可以配置任何你想要的系统和应用程序。这个体系结构两个最重要的需求是数据控制和数据捕获，其中数据控制尤为重要。无论数据控制机制多强力，但是却没有办法完全消除风险。下章我们将对主动防御安全系统的结构模型进行深入的研究。

3 主动防御安全系统结构模型研究

本章将深入分析入侵检测系统功能、利用 Snort 和 Base 实现安全分析审计系统，实际调研当前无线网络安全的状况及分析报告，介绍 Honeyd 及构建虚拟无线蜜网。

3.1 入侵检测系统模型的分析

3.1.1 入侵检测系统概述

1）入侵检测定义及分类

入侵检测系统(Intrusion Detective System)，入侵检测顾名思义，便是对入侵行为的发觉。它从网络或计算机系统中的若干关键点收集信息，并分析这些信息，检查网络中是否有违反安全策略的行为和遭到攻击的迹象。入侵检测被认为是防火墙之后的第二道安全闸门。入侵检测的主要任务是：

（1） 监视、分析用户及系统的活动。

（2） 系统构造和弱点的审计。

（3） 识别反映已知进攻的活动模式并向相关的系统进行警报。

（4） 异常行为模式的统计分析。

（5） 评估重要系统和数据文件的完整性。

（6） 操作系统的审计跟踪管理，并识别用户违反安全策略的行为。

入侵检测分为几类：简单模式匹配、状态模式匹配、基于协议解码的签名、启发式签名和异常检测（“签名”指一组条件， 如果满足这组条件的话，就表明是某种类型的入侵活动）[17]。

入侵检测系统实现有：HIDS(主机入侵检测系统)、NIDS(网络入侵检测系统)、Hybrid IDS（HIDS 与 NIDS 混合型入侵检测系统、DIDS（分布式入侵检测系统）。

2）信息捕获工具 Snort 特点

在部署的蜜网系统中三大核心技术之一的信息捕获中，我们使用了用来捕获入侵

者行为的 Snort 工具。Snort 是一个基于 libpcap 或 winpcap 的数据包臭探器，它具有截取网络数据报文，进行网络数据实时分析、报警以及日志记录等能力。它能检测各种不同的攻击方式，对攻击进行实时警报。此外，Snort 具有很好的扩展性和可移植性。Snort 是一个优秀开源软件的轻量级入侵检测系统，跟商业级的入侵检测系统相比毫不逊色，使用 Snort 的几大因素[18]：

（1）Snort 功能强大，基于 GPL 开放源代码让所有人都免费使用，方便移植到各种不同平台，支持所有的 Linux 系列，Solaris、HP-UX、IRIX、windows 等各种商业操作系统。

（2）Snort 具有实时流量分析和日志记录能力，能快速实时检测攻击并以各种方式作出警报。

（3）Snort 能够进行协议分析，内容的搜索/匹配。

（4）Snort 日志格式多样，支持 Tcpdump 二进制格式、ASCII 编码字符，还可以把日志记录到数据库中，支持最常见的数据库：Mysql、Postgresql、Mssql、UnixODBC 数据库、Oracle。下一节介绍的 base（安全基本分析引擎）就是利用 Snort 把日志记录到 mysql 数据库的。

（5）Snort 还有很强系统防护能力。我们蜜网系统 Honeywall 中的数据控制中的 Snort-Inline 就是用修改过的 Snort 与 Iptable 中的 rc.firewall 配合使用，进行阻塞或修改攻击请求。

（6）扩展性能较好，对于新的攻击威胁反应迅速。作为一个轻量级的网络入侵检测系统，Snort 有足够的扩展能力。它使用一种简单的规则描述语言(很多商用入侵检测系统都兼容 Snort 的规则语言)。最基本的规则知识包含四个域：处理动作，协议，方向，端口。例如：Alert tcp any any -> 192.168.0.0/24 23，清楚明了。发现新攻击后，可以很快地根据协议分析找到特征码，写出新的规则文件。

3.1.2 Snort 入侵检测模型分析

Snort 入侵检测是一个优秀入侵检测系统的标准，其开源代码让我们有机会深入

研究和学习入侵检测系统的内部框架及工作机理[19]。

1）Snort 入侵检测的结构原理分析

Snort 作为一个 NIDS（网络的入侵检测系统），其工作原理为在基于共享网络上检测原始的网络传输数据，通过分析捕获的数据包，主要工作为匹配入侵行为的特征或者从网络活动的角度检测异常行为，进而采取入侵的预警或记录。从检测模式而言，Snort 属于是误用检测（Misuse detection）[20]，误用检测就是对已知攻击的特征模式进行匹配，包括利用工作在网卡混杂模式下的嗅探器被动地进行协议分析，以及对一系 列数据包解释分析特征。从本质上上来说，Snort 是基于规则检测的入侵检测工具，即 针对每一种入侵行为，都提炼出它的特征值并按照规范写成检验规则，从而形成一个规 则数据库。其次将捕获得数据包按照规则库逐一匹配，若匹配成功，则认为该入侵行为 成立。Snort 的结构主要分为三个部分：

（1）数据包捕获(Capture Packet Mechanism from link layer)

该子系统的功能为捕获网络得传输数据并按照 TCP/IP 协议的不同层次将数据包进行解析。Snort 开始工作时，先把网卡设置为混杂模式，这样可以接收本地局域网中所有的封包，初始化封包捕获模块（Uinx 或 Linux 系统加载 LibPcap 模块，Windows 则调用 winpcap），Snort 利用 libpcap 库函数进行采集数据，该库函数可以为应用程序提供直接从链路层 捕获数据包的接口函数并可以设置数据包的过滤器以来捕获指定的数据[15]。

（2）解析子系统(Packet decoder)

数据包捕获后调用数据包协议解码模块，并将解码结果存储在特定的 Packet 类型数据结构。网络数据采集和解析机制是整个 NIDS 实现的基础，其中最关键的是要保 证高速和低的丢包率，这不仅仅取决于软件的效率还同硬件的处理能力相关。对于解析机制来说，能够处理数据包的类型的多样性也同样非常重要，目前，Snort 可以处理以太 网(图 3-1 以太网封包解码、图 3-2ARP 数据包解码序列图)，令牌环以及 SLIP 等多种链路类型的包。

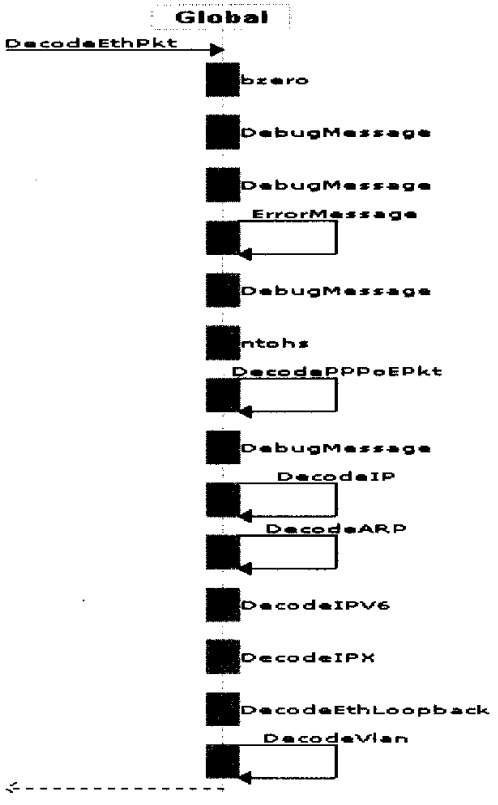


图 3-1　以太网封包解码序列图

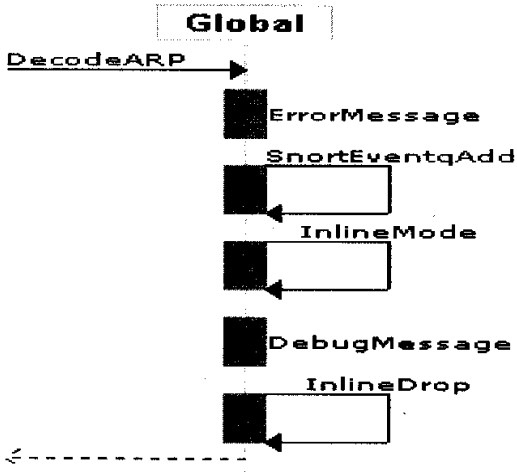


图 3-2　ARP 数据包解码序列图

（3）检测引擎（the detect engine）

检测引擎是 Snort 实现网络入侵检测的核心，准确性和快速性是衡量其性能的重要指标，前者主要取决于对入侵行为特征码的提炼的精确性和规则撰写的简洁实用性，由于网络入侵检测系统自身角色的被动性——只能被动的检测流经本网络的数据，而不能主动发送数据包去探测，所以只有将入侵行为的特征码归结为协议的不同字段的特征值，通过检测该特征值来决定入侵行为是否发生。后者主要取决于引擎的组织结构，是否能够快速地进行规则匹配[21]。

2）Snort 系统模块分析

Snort 入侵检测系统的工作总流程图可以参看（图 3-3）

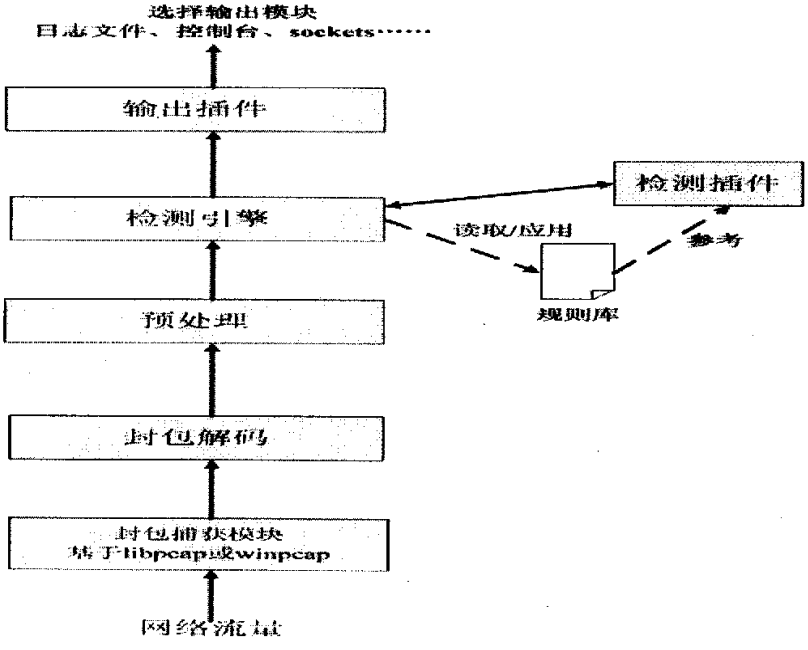


图 3-3 Snort 入侵检测的工作总流程图

Snort 入侵检测由 Snort 主函数模块及各功能模块组成。Snort 主模块与各功能模块之间的关系序列图如图 3-4，从图中看到之间的如何工作流程分析如下[22]：

（1）命令行参数解析函数 ParseCmdLine()，解析个中命令参数，并将其放入全局 PV 类型的变量 PV 中。数据类型 PV 包含各种标示字段，用来知识个中系统的参数设置。例如：规则文件，系统运行模式，显示模式，插件激活等。

（2）检测引擎初始化 fpInitDetectinEngine(),用于制定快速规则匹配模块的配置参数，包括模式搜索算法等(Snort 可使用的算法有 Aho-Corasick ，Wu-Manber ，Boyer-Moore 等算法,缺省 Snort 使用 Byer-More 算法。)。并负责在协议解析过程中产生警报信息。

（3）取得从网络截取到的数据流的主要进程 OpenPcap()。起主要作用是根据命令行参数分析结果，分别调用 Libpcap 函数库 Pcap_open_live()—通过网卡驱动实时截取网络数据 和 Pcap_open_offline()—通过文件来访问以前网卡驱动截取数据保存成为的文件，并获得相对应的数据包数据结构。

（4）各种插件初始化主要包括输入/输出插件初始化，检测插件初始化和预处理插件初始化等。主要就是将各种插件的关键字与对应的初始化处理函数相调用，然后注册到对应的关键字链表结构中，随后逐个弹出以便规则解析模块使用[23]。

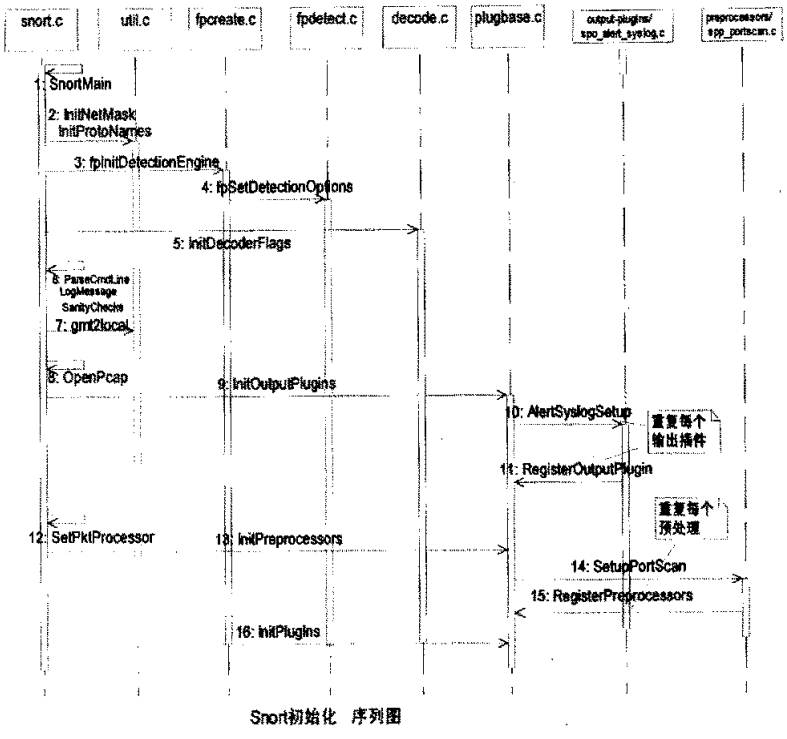


图 3-4 snort 初始化序列图

3.2 Snort 和 Base 入侵检测系统设计及实现

随着黑客技术发展及攻击工具的自动化，分布式环境下的协同攻击与日俱增，然而传统的 IDS 局限于单一主机或网络架构，不同的 IDS 系统之间不能协同工作。为解决这一问题，多级互动的分布式协同入侵检测技术与通用入侵检测架构应运而生。最先出现的分布式 IDS 利用分布在网络中的传感器扩大数据源的范围，但最终的数据分析却是集中进行的。这样做带来的问题有两个：一是整个系统有一个中心控制点，该点的失效将导致整个系统失效；二是集中进行数据分析使负载集中在承担分析工作的主机上，限制了系统的可扩展性与效率的提高[24]。

BASE(安全基本分析引擎)是一个基于 PHP 的分析引擎，用于分析、查询、管理 Snort 传感器产生的数据库。它来源于入侵数据库分析控制台 ACID (Analysis Console for Intrusion Databases)[17]。BASE 具有搜索、分组、维护和图示数据库的日志数据。这些数据既可以是 Snort 的日志/警报，也可以是其它防火墙产生的日志信息。本文提供一种基于 Snort+Base 的分布式协同入侵检测系统的设计框架，单个传感器是基于 Snort 的，本身具有数据采集，入侵分析、响应的功能；单个传感器把分析结果上传给所属域的基于 php 的控制台 BASE，由控制台统一管理警告信息，并图形化分类显示，这个域的管理员可以根据当时的情况和网络需求在控制台远程控制各个传感器(Sensor)。控制台和传感器之间传送的控制消息用 SSH 加密传输，避免数据传输过程中被嗅探，这样一来，传感器之间既具有独立性，并且通过远程配置，还可以使各个传感器之间安全地进行最大可能的分工协作。组件本身的功能不再是影响整个系统的关键因素，也就是说，一个组件失效仅仅使整个系统的能力降低，而不会使整个系统失效。系统的结构相对固定，而系统各组件何时该独立工作，何时需要协作则相对灵活。通过灵活分配角色的协作机制，能够更准确地反映出分布式各个组件之间的关系，具有更大的适用性[25]。

3.2.1 Snort 和 Base 的入侵事件数据库结构分析

1）Snort 入侵事件数据库结构

入侵事件数据库的作用是把Honeypot系统实时检测到的所有入侵事件记录下来，以便于管理员事后对其进行分析，从而得出关于当前黑客技术的相关报告，并在此基础上改进现有的系统和网络设置，为维护网络安全提供重要信息。Base入侵事件数据库是由16张标准的Snort二维表和3张Snort扩展表组成，Sensor是传感探测器表，envent是存放入侵检测事件表。详细表内容及关系可参看图3-6[23]。

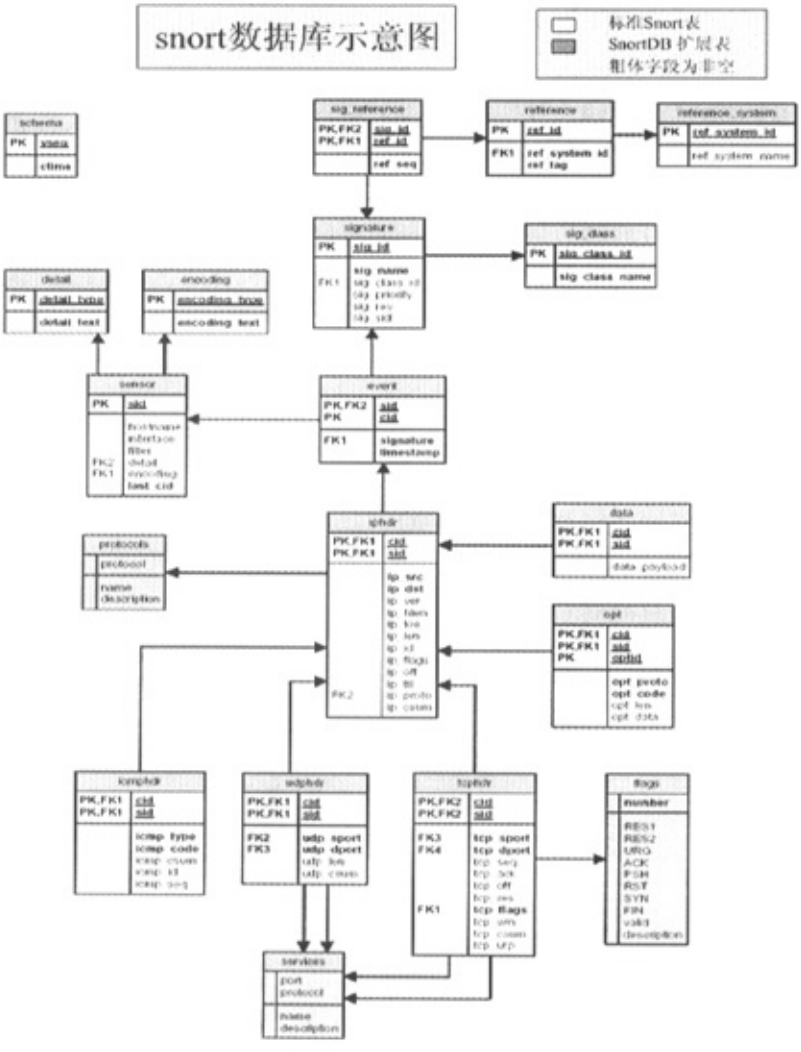


图3-6 snort入侵事件数据库关系示意图

从图中可以清楚地看到 Snort 数据封包的编码格式，基本是由 IP 数据报组成，IP 协议是 TCP/IP 协议族的最核心部分，所有的 TCP、UDP、ICMP、IGMP 都被封装在 IP 数据报中传送。IP 数据报的报头格式如图 3-7，与 Snort 定义的 IPHDR 数据表（如图 3-9）完全一致。TCP 的头部结构(如图 3-8)也基本与 TCPHDR 数据表（如图 3-10）一致。

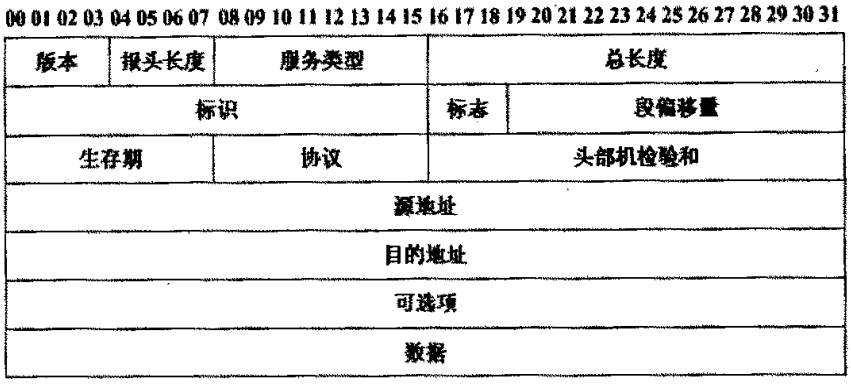


图 3-7　IP 头部格式

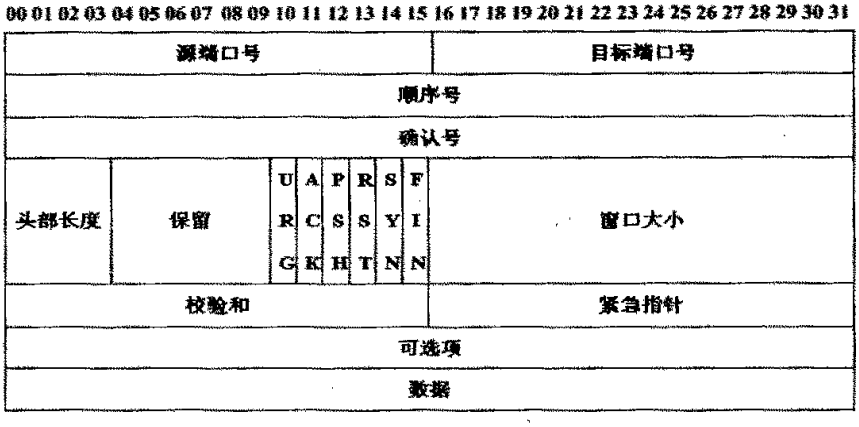


图 3-8　TCP 头部结构

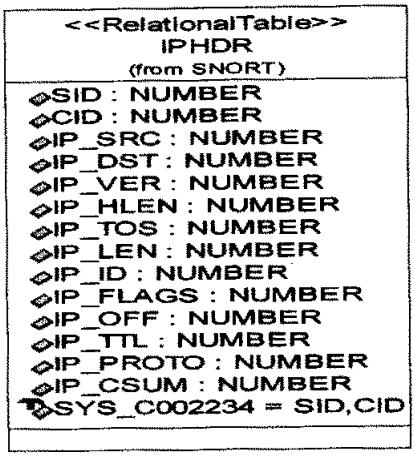


图 3-9 Snort 日志数据表 IPHDR

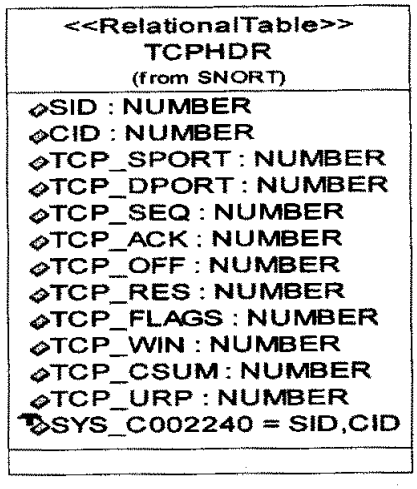


图 3-10 Snort 日志数据表 TCPHDR

通过分析 Snort 的日志存储数据库结构，可以有助于理解我们的安全基本分析引擎（BASE）是如何调用 Snort 数据库来进行入侵检测分析及报警的。

2）安全基本分析引擎 BASE 中 Snort 日志数据库结构分析

Snort 将收集到的信息存储到数据库中，安全基本分析引擎要将数据库中的信息根据查询端的要求，BASE 系统通过对数据库中历史数据的深入分析，使用系统预设

的各种模板，能够从不同角度（按网络设备、系统、事件类型等）生成日志分析结果，以直观、清晰的可视化图表予以显示。分析涵盖了对事件的归类统计及事件的变化发展趋势[27]。

本系统中的基本安全分析引擎 BASE 与入侵检测系统 Snort2.33 的日志数据库结构关系，可以从图 3-11 中清楚了解到。Snort 从网络中收集日志及监视数据包，然后解码及格式化 Snort 的编码格式，包括 IP、TCP、UDP、ICMP 等，具体编码格式可参看上一节。过滤不同的事件，然后与规则库中的签名进行比较，通过对事件的智能检测，最后生成分析结果，如果用户通过 WEB 接口进行查询，可以生成各种直观图形报表。

BASE(安全基本分析引擎)+SNORT的数据库关系图

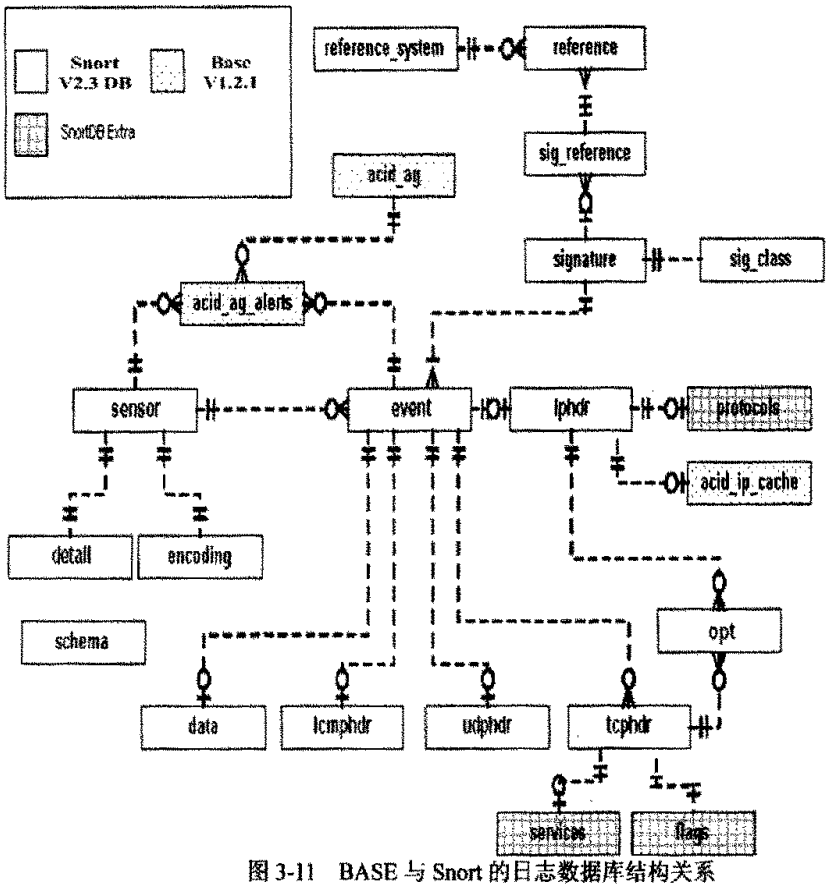


图 3-11 BASE 与 Snort 的日志数据库结构关系

3.2.3 Window 下 Snort 和 Base 安装部署

为了更好的对蜜网系统及入侵检测收集到的数据进行的分析，我们将部署基于 Windows 环境下的 Snort 结合 Base 的安全基本分析引擎审计系统[28]。

1）安装它们时所需要的组件以及它们的作用，功能（各个安装文件去相关软件的主页下载）：

（1） WinPcap：Windwos 下的捕获网络数据包的驱动程序库，http://winpcap.polito.it/

（2）Snort：将其捕获的数据发送至数据库，http://www.snort.org/

（3）Apache：为系统提供了 web 服务支持，http://www.apache.org/。

（4）Php：为系统提供了 php 支持，使 apache 能够运行 php 程序，http://www.php.net/。

（5）MySQL：存储网络数据包的数据库，http://www.mysql.com/。

（6）Base：基于 php 的入侵检测数据库安全基本分析引擎（刚才安装 apache 和 php 就是为了能运行它）http://sourceforge.net/projects/secureideas 。

（7）Adodb：php 数据库的连接组件，http://adodb.sourceforge.net/。

（8） Phplot：图形链接库 For PHP，http://www.sourceforge.net/projects/phplot/。

Base：通过 adodb 从 mssql.snort 数据库中读取数据，将分析结果显示在网页上，并对其进行图形化分析。

2）安装步骤

（1）安装 WinPcap。

（2）安装 Apache，PHP，MySQL，apache 使用带 ssl 认证版本的 Apache_2.0.55-Openssl_0.9.8a-Win32，Apache 安装到 c:\apache2，安装完后修改 apache2\conf\httpd.conf 中 servername 的成你自己 IP 地址或域名，加一行 LoadModule php5_module c:/php/php5apache2.dll 以对 php 模块支持，加入 AddType application/x-httpd-php .php，DirectoryIndex index.html index.php，运行“apache2\bin\apache -k install”命令安装 apache 作 windows 2000 的服务启动。将 php

目录下的 php.ini 拷贝到%systemroot%目录，将 php 目录下 libmysql.dll、ntwdblib.dll、php5apache2.dll 拷贝到%systemroot%\system32 目录下，同时要加载 GD 库。安装 mysql 作 windows 2000 的服务启动，往 mysql 添加 snort 用户，并设置相应权限。

（3）安装 Snort

使用默认安装路径 c:\Snort，选择数据库为 MySQL，并按照以下修改 C:\Snort\etc\snort.conf 文件：

var RULE_PATH c:\snort\rules

output database: log, mysql, user=root password= dbname=snort host=localhost

output database: alert, mysql, user=root password= dbname=snort host=localhost

include C:\Snort\etc\classification.config

include C:\Snort\etc\reference.config

安装 snort 作为 windows 服务，转到 c:\snort\bin 运行 snort /SERVICE /INSTALL -c "c:\snort\etc\snort.conf" -l "c:\snort\logs" -d

（4）安装 adodb 和 phplot

分别解压缩 adodb461.zip、phplot-5.0rc2.tar.gz 至 \apache2\htdocs\adodb 及 \apache2\htdocs\phplot 目录下

（5）安装 Base

① 解压缩 Base 压缩包至 apache2\htdocs\Base 目录下

② 修改 base_conf.php 文件，找到相应的行，并把它们改成：

$BASE_Language = "chinesegb";//base 语言

$DBlib_path = "c:\apache2\htdocs\adodb";

$ChartLib_path = "c:\apache2\htdocs\phplot";

$portscan_file = "c:\Snort\log\portscan.log";

$DBtype = "mysql"; //日志数据库类型

$alert_dbname = "snort";

$alert_host = "localhost";

$alert_port = "3306";

$alert_user = "snort";

$alert_password = "snort_user_password"; //mysql 中 snort 用户密码

③ 打开 http://localhost/base/base_db_setup.php，测试基本功能是否安装成功。如果有错误，则根据错误情况重新检查。在正常情况下，到此处应该能够正常连接数据库。

3）改进 BASE 对中文支持

Base 对中文支持不是很好，特别是在生成图形报表时会出现乱码，对 BASE 在中文处理方面作了如下改进，先由 PHP 生成 GBK 中文字库，然后再在图形处理程序中调用 GBK 字库，具体程序代码见附录 A。

4）Snort 和 BASE 的运行结果：运行 Snort，开始运行中输入 net start snort，输入 http://localhost/base/，应该可以看到当前的网络情况分析（图 3-12）及 Snort 警告的详细说明（图 3-13）。

图 3-12　安全基本分析引擎的首页

图 3-13　Snort 警告的详细说明

BASE 可有根据时间、IP 地址来源、TCP 及 UDP 端口等的警告生成条形、线形、圆形不同类别的图形报表。图 3-14 是根据源 UDP 端口警告生成的图形报表。

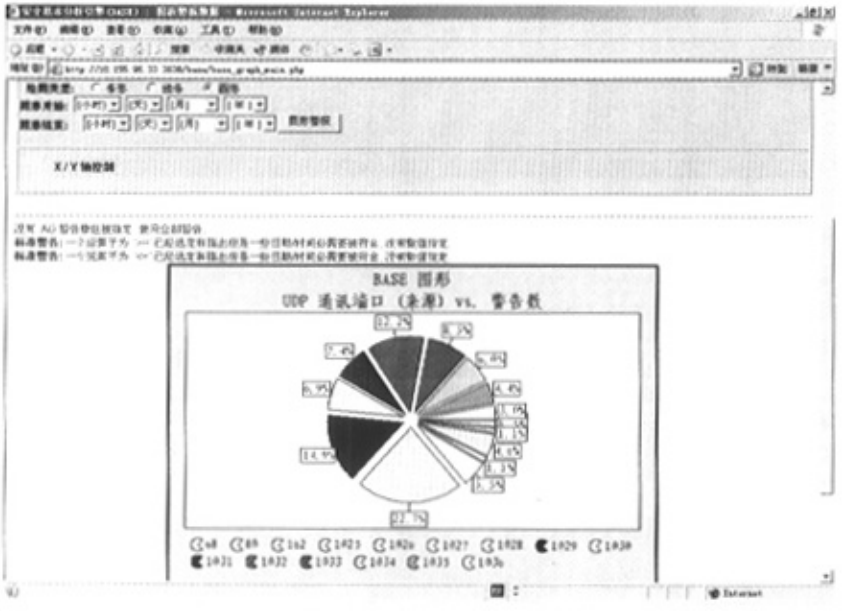


图 3-14　源 UDP 端口生成的报表图

3.3 无线蜜网系统的部署

3.3.1 无线局域网的安全状况

最近几年，无线技术迅猛发展，现在无论在公司还是家庭都在广泛使用。人们已经对那些技术产生了很大的依赖性，以至于无线设备随处可见，从网络设备到笔记本电脑、照相机等等。 虽然这些设备支持标准的安全选项和协议来对付普通攻击（加密、验证等），但是由于真实使用的安全等级以及攻击者掌握的技术水平，所以这些设备还是可能受到攻击。 有时，黑帽子发现公司无线网络非常脆弱甚至没有安全性，他们能够以此为撕破口，进一步渗透到内部的机器来窃取信息或者作为跳板攻击 Internet 上其它机器以隐藏自己的踪迹。这些威胁可以穿越外部物理屏障（比如来自停车场，公司下的街道、窗口）或者就在你的内部环境，因为攻击者使用手持 PDA 或笔记本电脑，加上无线网卡和扫描软件就可以发现网络[29]。

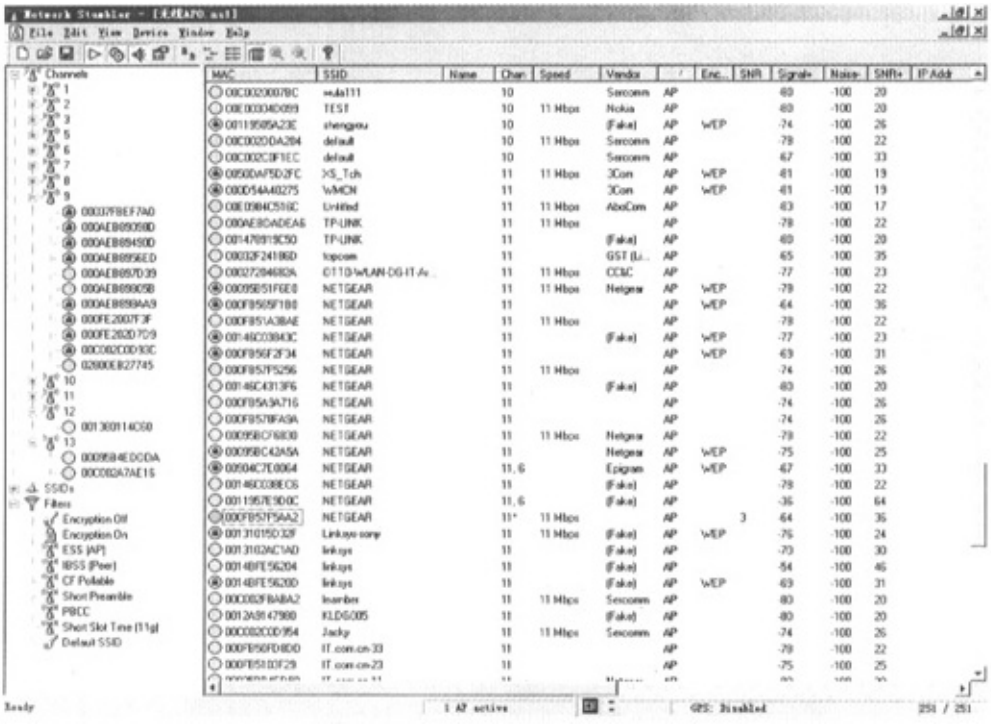


图 3-15 NetStumbler 扫描到的无线 AP

在无线网络世界中，黑客组织入侵无线网络的技术叫“战争驾驶”(War Driving)，这种入侵方式很简单，黑客携带配有 802.11 无线网卡的笔记本电脑或 PDA ，开车沿街驾驶试图找到、识别并渗透无线网络。成功访问您的网络的驾驶攻击者可以匿名劫持您的 Internet 连接、窃取存储在在您的网络上的个人信息、截取文件传输，或者甚至将您的计算机用作一种“僵尸”以发送可以追溯到您的垃圾邮件或恶意软件。图 3-15 显示使用 NetStumbler 扫描到的无线 AP。

表 3.1 用户使用无线网络安全情况表

AP 总数	使用 WEP 加密	无 WEP 加密	修改 SSID	使用产品默认 SSID	同时 WEP 加密及修改默认 SSID
251	96	155	128	123	31
百分比	38．25%	61.75%	51%	49%	12.35%

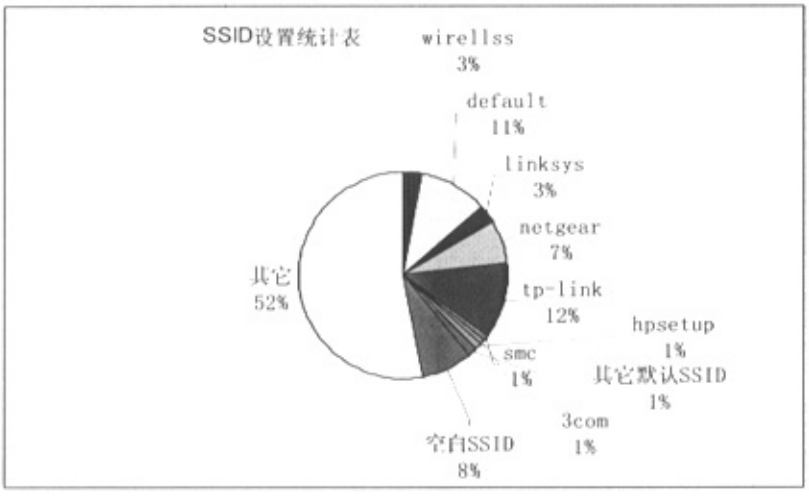


图 3-16 无线网络 AP 设置为出厂默认 SSID 统计表

无线网络安全的威胁一是由于 802.11 本身的安全性很脆弱，别一重要的方面是源于用户缺少安全保密的意识。无线安全由 SSID(System Set ID－系统集合标识名)和 WEP(Wired Equivalent Privacy－有线对等保密)构成认证和加密两个重要的安全门槛，简单的说可以把 SSID 比喻为一个房间的大门钥匙，而 WEP 即是房间里的保险柜的密码，黑客有了 SSID 这把钥匙就可以进入你网络，但如果使用了 WEP 加密黑客就不能

通过网络嗅探直接看到数据，因为数据被用 WEP 加密锁在保险柜里。根据我的一次调查扫描结果的统计显示，在扫描调查中发现 251 个无线 AP，使用 WEP 加密的有 96 个，但同时使用 WEP 加密修改及修改 AP 出厂设置 SSID 值用户数只有 31 个，61%的没有大部分没有使用 WEP 加密，49%无线 AP 使用的都是出厂默认的 SSID。具体可参见表 3.1 和图 3-16。从这些数据中我们可以看到无线安全主要的还是用户的安全意识浅薄。这也给有意入侵无线网络的黑客大开方便之门。

3.3.2 构建无线蜜网的价值

构建无线蜜罐有何价值？这取决于研究者的网络和对安全的需要。当研究者使用无线蜜罐后，研究者会对它产生的好处感兴趣的。在黑客团体里，有很多人喜欢攻击无线网络，因为对无线网络的攻击有如下几个特点[30]：

（1）安全：在攻击的时候无需实际的物理连接，可以保持较远的距离，所以被发现的时候可以跑开；

（2）简单：现在到处都有大量开放或者不安全的 AP（酒店、机场、公共区域无线服务点，还有 SOHO 无线网络等）。这种设备越来越便宜，而且数量越来越多；

（3）技术新：无线网络的攻击比较有吸引力和挑战性；

（4）隐蔽：对于黑客和网络恐怖分子来说，无线网络是一个理想的作案平台。随机使用开放 AP 连接网络，增加了攻击的隐蔽性，被捉获的可能性很小。

但是很多网络管理员常常认为，针对无线的攻击使得攻击者在位置上必须靠近网络设备，所以无线攻击的危险性要小于 Internet 的攻击。而且他们坚信这些攻击很少发生。但是对许多公司来说，无线网络被攻破带来的后果是非常严重的。无线蜜罐可以帮助你的网络得到受攻击的真实统计数据，包括攻击频率、攻击者的技术水平、攻击得手次数以及使用的方法。无线蜜罐同样可以保护你的网络，因为它的迷惑性使得攻击者花费大量精力对付伪造的目标，而且不易发现你网络的真实构架[34]。

3.3.3 无线网络的数据加密算法

无线入侵者针对无线网络技术中涉及的安全认证加密协议的攻击与破解就层出

不穷。无线加密最初使用的是 WEP一有线对等保密算法进行数据加密，但 WEP 本算存在很多弊端，很快出现另一过临时过渡无线数据加密算法 WAP(Wi-Fi Protected Access)[31]。

1）WEP 的加密算法

无线网络设计时为了使无线业务应用能达到与有线业务同样的安全等级，IEEE802.11 标准中采用了 WEP 协议来设置专门的安全机制。WEP 采用对称加密机制。加密和解密使用相同的密钥和加密算法。启用 WEP 加密后，必须两端设备同时启用相同的加密密钥才能进行通信，否则无法通信。

（1）WEP 加密过程

WEP 支持 64 位和 128 位加密，对于 64 位加密，加密密钥为 10 个十六进制字符(0-9 和 A-F)或 5 个 ASCII 字符(图 3-17);对于 128 位加密，加密密钥使用 26 个十六进制字符或 13 个 ASCII 字符。

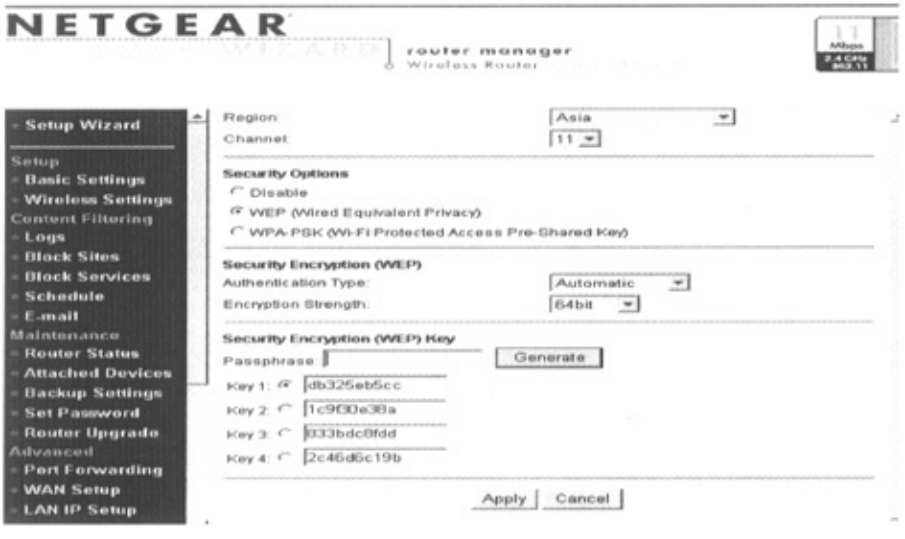


图 3-17　64 位加密密钥

WEP 依赖通信双方共享的密钥来保护所传的加密数据及其数据的加密过程如下。

①计算校验和(Check Summing)。

a. 对输入数据进行完整性校验和计算。

b. 把输入数据和计算得到的校验和组合起来得到新的加密数据，也称之为明文，明文作为下一步加密过程的输入。

②加密。在这个过程中，将第一步得到的数据明文采用算法加密。对明文的加密有两层含义:明文数据的加密，保护未经认证的数据。

a. 将 24 位的初始化向量和 40 位的密钥连接进行校验和计算，得到 64 位的数据。

b. 将这个 64 位的数据输入到虚拟随机数产生器中，它对初始化向量和密钥的校验和计算值进行加密计算。

c. 经过校验和计算的明文与虚拟随机数产生器的输出密钥流进行按位异或运算得到加密后的信息，即密文。

③传输。将初始化向量和密文串接起来，得到要传输的加密数据帧，在无线链路上传输(如图 3-18 所示)。

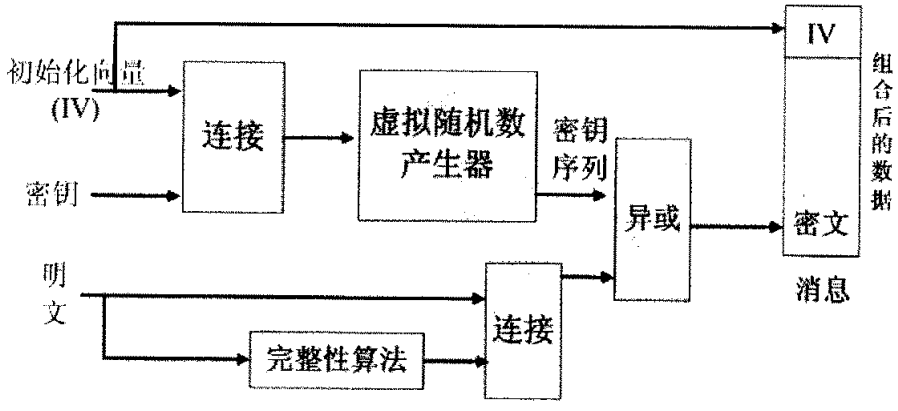


图 3-18 WEP 加密过程

（2）WEP 解密过程

在安全机制中，加密数据帧的解密过程只是加密过程的简单取反。解密过程如下。

①恢复初始明文。重新产生密钥流，将其与接收到的密文信息进行异或运算，以恢复初始明文信息。

②检验校验和。接收方根据恢复的明文信息来检验校验和，将恢复的明文信息分离，重新计算校验和并检查它是否与接收到的校验和相匹配。这样可以保证只有正确

校验和的数据帧才会被接收方接受。如图 3-19 所示

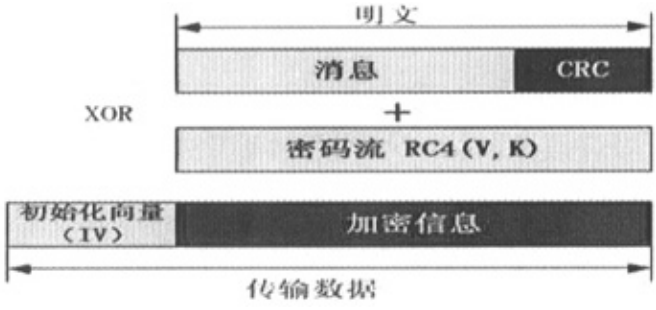


图 3-19 WEP 解密过程

2）WPA 加密算法

无线数据加密的 WEP 其存在很多致合的弱点，导致其很容易被黑客所破解。因而必须找到另一种加密算法，WPA 是就是这个算法中的一个过渡标准。WPA 将临时密钥完整性协议 (TKIP) 与 Michael 结合起来，取代了有线对等保密 (WEP)；临时密钥完整性协议可通过加密来保证数据机密性，Michael 可保证数据完整性[32]。

（1）WEP 的弊端及 WPA 解决对策

802.11 标准中的 WEP 在加密方面存在以下弊端：

• 初始化向量 (IV) 太小。

WEP 将 IV 及 WEP 加密密钥用作 RC4 伪随机数生成器 (PRNG) 的输入，生成用来加密 802.11 帧有效负载的密钥流。有了 24 位的 WEP IV 之后，就很容易捕获多个具有相同 IV 值的 WEP 帧，从而使实时解密更加容易。

• 弱数据完整性

WEP 数据完整性包括对非加密 802.11 有效负载中的字节执行循环冗余校验 32 (CRC-32) 校验和计算，然后使用 WEP 对它的值进行加密。即使在加密后，也比较容易更改加密有效负载中的位并适当更新加密 CRC-32 结果，从而阻止接收节点检测到帧内容已发生更改这一情况。

• 使用主密钥而不使用派生密钥

WEP 加密密钥（或者是手动配置的，或者是通过 802.1X 身份验证确定的）是唯一可用的密钥材料。因此，WEP 加密密钥是主密钥。使用主密钥加密数据不如使用从主密钥派生的密钥安全。

• 不重新生成密钥

WEP 没有提供刷新加密密钥的方法。

• 无重放保护

WEP 不能防范重放攻击。在重放攻击中，攻击者会发送一系列的以前捕获的帧，试图以此方式获得访问权或修改数据。

为了能克服以 WEP 以上弊端，WPA 作了以下的改进（见表 3.2）

表 3.2 WEP 的弊端及 WPA 解决对策

WEP 的弊端	WPA 解决方法
IV 太短	在 TKIP 中，IV 的大小增加了一倍，已达 48 位。
弱数据完整性	WEP 加密的 CRC-32 校验和计算已经由 Michael 取代，Michael 是一种专门用于提供强数据完整性的算法。Michael 算法可以计算 64 位消息完整性代码 (MIC) 值，该值是用 TKIP 加密的。
使用主密钥，而不使用派生密钥	TKIP 和 Michael 使用一组从主密钥和其他值派生的临时密钥。主密钥是从“可扩展身份验证协议-传输层安全性”(EAP-TLS) 或受保护的 EAP (PEAP) 802.1X 身份验证过程派生而来的。此外，RC4 PRNG 的输入的机密部分是通过数据包混合函数计算出来的，它会随着帧的改变而改变。
不重新生成密钥	WPA 自动重新生成密钥以派生新的临时密钥组。
无重放保护	TKIP 将 IV 用作帧计数器以提供重放保护。

（2）WPA 加密和解密过程

WPA 需要使用下列值来为无线数据帧提供加密和完整性保护：

IV，以 0 开始，随每个后续帧而递增

- 数据加密密钥（用于单播通信量）或组加密密钥（用于多播或广播通信量）
- 无线帧的目标地址（DA） 和源地址（SA）
- 一个优先级字段的值，被设置为 0，保留以备以后使用
- 数据完整性密钥（用于单播通信量）或组完整性密钥（用于多播或广播通信量）

下图 3-20 说明单播数据帧的 WPA 加密过程。

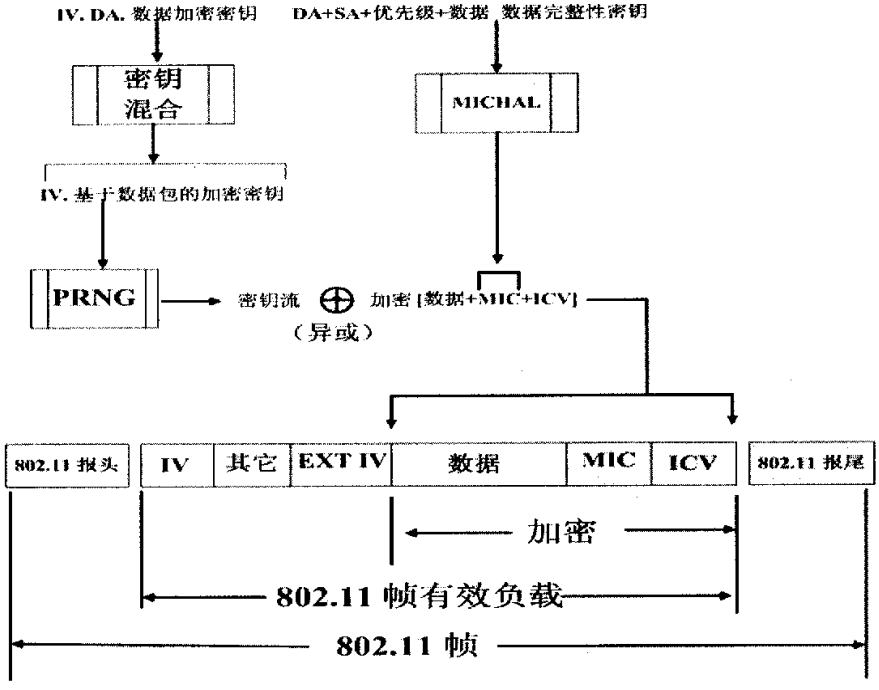


图 3-20 WPA 加密过程

加密过程说明

①IV、DA 和数据加密密钥被输入 WPA 密钥混合函数，该函数计算基于每个数据包的加密密钥。

②DA、SA、优先级、数据（非加密 802.11 有效负载）和数据完整性密钥被输入 Michael 数据完整性算法以生成 MIC。

③ICV 是从 CRC-32 校验和计算出来的。

④IV 和基于每个数据包的加密密钥被输入 RC4 PRNG 函数以生成与数据、MIC 和 ICV 大小相同的密钥流。

⑤密钥流与数据、MIC 和 ICV 的组合进行异或逻辑运算，生成 802.11 有效负载的加密部分。

⑥IV 被添加到 IV 和扩展 IV 两个字段中的 802.11 有效负载的加密部分，其结果被 802.11 报头和报尾封装了起来。

图 3-21 说明单播数据帧的 WPA 解密过程

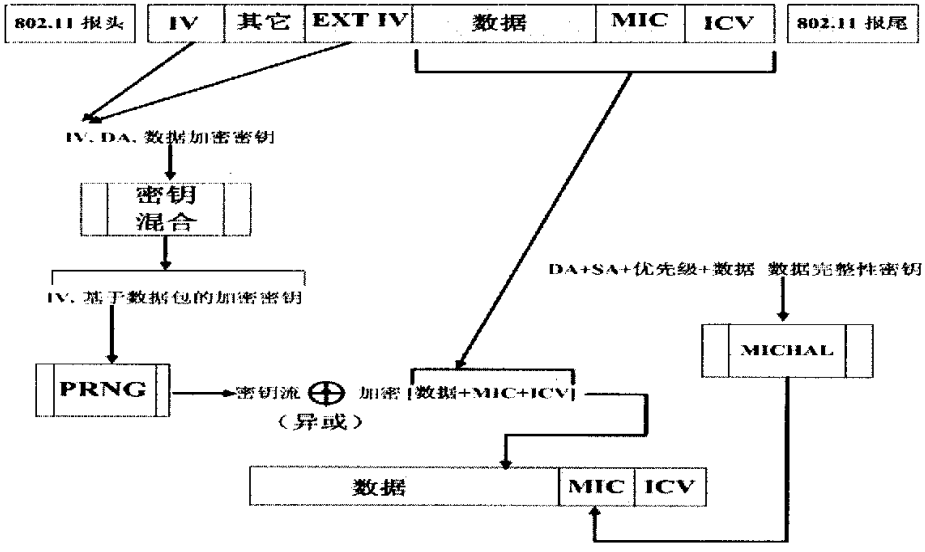


图 3-21　WPA 解密过程

WPA 解密过程说明如下：

①从 802.11 帧有效负载的 IV 和扩展 IV 两个字段中提取 IV 值，然后将此值与 DA 和数据加密密钥一起输入密钥混合函数，生成基于数据包的加密密钥。

②IV 和基于数据包的加密密钥被输入 RC4 PRNG 函数，生成与加密的数据、MIC 和 ICV 大小相同的密钥流。

③密钥流与加密的数据、MIC 和 ICV 进行异或逻辑运算，生成非加密数据、

MIC 和 ICV。

④计算 ICV，并将其与非加密 ICV 的值相比较。如果两个 ICV 值不匹配，数据就会被悄悄丢弃。

⑤DA、SA、数据和数据完整性密钥被输入 Michael 完整性算法以生成 MIC。

⑥MIC 的计算值与非加密 MIC 的值相比较。如果两个 MIC 值不匹配，数据就会被悄悄丢弃。如果两个 MIC 值相匹配，数据就会被传输到上一级网络层进行处理[30]。

3.3.4 无线蜜网的部署

为了成完无线蜜网的部署，我们最少必须有一台提供无线接入的设备，这个设备可以是真实的 AP,也可以是虚拟的 AP,如果是真实的 AP,把它接入到有线网络中，做好一定的安全防范，提供一定的可见有用的资源作诱饵吸引攻击者，同时还要有一个进行隐蔽的数据捕获客户端。为了监视二层无线攻击，我们可以在处于监视模式的无线客户端上运行 Kismet 软件实现此目的。图 3-22 展示一个无线蜜罐的拓扑结构图[33]。

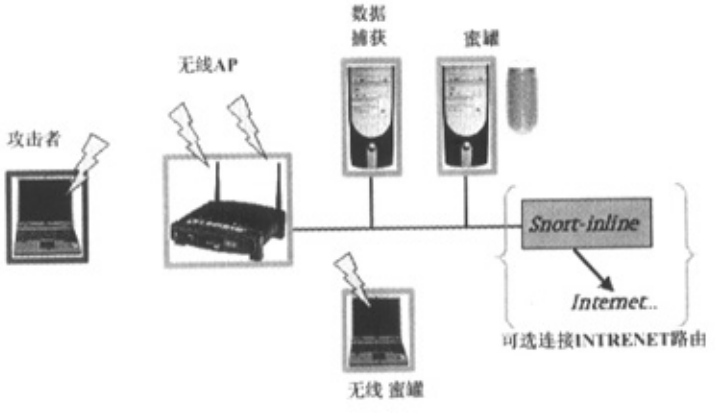


图 3-22 无线蜜罐的拓扑结构图

无线蜜罐如果接入 Internet，必须使用入侵防御系统阻止攻击者通过你的蜜罐作跳板实现对外网络传输的攻击。可以使用蜜网项目组织的 snort-inline 来实现此功能。

无线蜜罐中的构建，除了使用真实的 AP 外，另一简单的解决方案：可以把一块无线网卡设置成 Master 模式并运行就可以把蜜罐系统搭建起来，这方案简单实用。不连接实际网络就算被攻被也不会危及到其他系统。而蜜罐的交互性也非常容易调整。

1）利用 Honeyd 构建无线蜜罐系统

（1）Honeyd 的介绍

Honeyd 是一个很酷很小巧的用于创建虚拟的网络上的主机的后台程序，这些虚拟主机可以配置使得它们提供任意的服务，利用个性处理可以使得这些主机显示为在某个特定版本的操作系统上运行。Honeyd 可以根据 nmap 的指纹来模拟任意多个操作系统和网络服务，支持 IP 协议族，可以创建任意拓扑结构的虚拟网络。Honeyd 是基于 GPL 下的开源软件,但很多商业公司也在使用它。Honeyd 可以通过提供威胁检测与评估机制来提高计算机系统的安全性，也可以通过将真实系统隐藏在虚拟系统中来阻止外来的攻击者。因为 Honeyd 只能进行网络级的模拟，不能提供真实的交互环境，能获取的有价值的攻击者的信息比较有限，所以常常用 Honeyd 模拟蜜罐系统来转移攻击者的目标[34]。

（2）Honeyd 数据收集

Honeyd 被设计用来应答目标地址属于模拟蜜罐范围之内的网络包。要让 Honeyd 能够接受到发送给虚拟蜜罐的数据包，必须要正确的配置网络。有几种方法可以实现网络的配置，如为指向 Honeyd 主机的虚拟 IP 建立特殊的路由、使用代理 ARP（地址解析协议）或者使用网络隧道等。最常见的代理 ARP 软件有 linux 下的 arpd,debain 系统为 farpd。

（3）Honeyd 的体系结构

Honeyd 体系由几个组件构成，这些组件是配置数据库、中央包分发器、协议处理器、个性引擎和可选路由构件。如下图 3-23 所示。

我们来一步了解 honeyd 的工作原理，系统接受到的数据会由中央包分发器进行处理，首先中央包分发器进行处理会检查 IP 包的长度，修改包的校验和。Honeyd 框架响应的是最主要的 3 种互联网协议：ICMP、TCP 和 UDP，其他协议的包在被记入日志后会被悄悄丢弃。

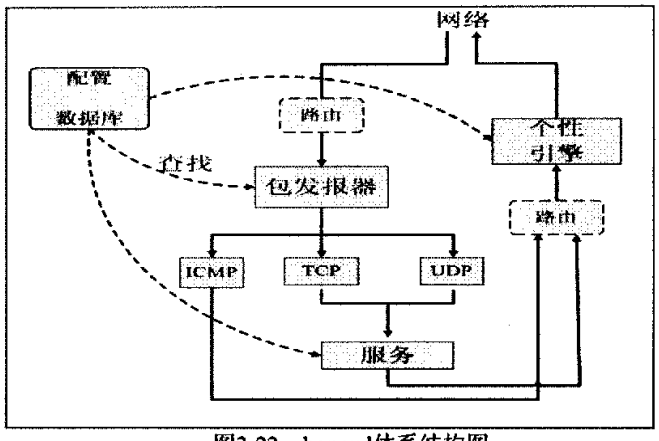


图3-23 honeyd体系结构图

不同操作系统的网络栈处理各不相同，这导致他们所发送的数据包具有各自不同的特点，这是每个系统的指纹特殊“个性”。Honeyd 为了能更真实的虚拟不同的操作系统和网络服务而借助 nmap 和 xprobe 的指纹识别来实现虚拟蜜罐的“个性”。图 3-24 是 Honeyd 的 UML 类图，其中的 personality 类就是实现“个性引擎”的功能模块。

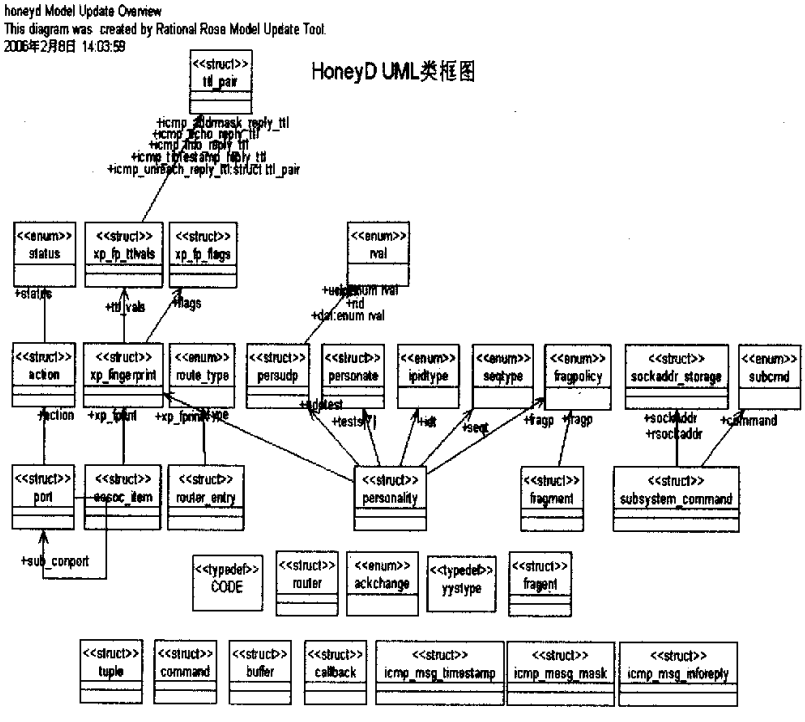


图3-24 Honeyd的UML类图

2）honeyd 虚拟的无线蜜罐

虚拟蜜罐的配置比较简单，下面我们利用 honeyd 来虚拟配置一台具有 web 管理功能的 Linksys WRT54 AP，具体操作如下[32]：

（1）我们先将网卡设置为 master 模式，

#iwconfig wlan0 mode master

（2）修改配置 honeyd.conf 文件，加入如下脚本

Create linksys

#下面是设置 linksys 指纹个性，这款无线路由 ap 建于 LINUX 操作系统

Set linksys personality “Linux Kernel 2.4.0 – 2.5.20”

add linksys tcp port 80 “/usr/share/honeyd/scripts/fakelinksys.sh”

add linksys udp open 53 open

add linksys udp open 67 open

add linksys udp open 69 open

set linksys tcp action reset

set linksys uid 65534 gid 65534

bind 10.155.99.1 linksys

无线攻击者使用 nmap –O（检查系统指纹参数）将会看到如下内容：

Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20

对于保持蜜罐活动性来说无线网络模拟数据传输非常重要，因为攻击者往往需要看到有数据传输才可以实施他们的攻击。我们可以通过脚本的自动应答和执行命令这个功能来实现模拟无线数据的传输。为了能让虚拟无线蜜罐回应攻击者的入侵，Honeyd 需要使用 fakelinksys.sh 脚本来处理 Web 请求，fakelinksys.sh 脚本如下：

```
    #!/bin/sh
DATE='date'
echo "== Httpd break-in attempt [$DATE] ==" >> /tmp/linksys.log
while read request
do
  LINE='echo "$request" | egrep -i "[a-z:]"'
  if [ -z "$LINE" ]
```

```
  then
    break
  fi
  echo "$request" >> /tmp/linksys.log
done
echo "==" >> /tmp/linksys.log
    cat << _EOF_
HTTP/1.0 401 Unauthorized
Server: httpd
Date: $DATE
WWW-Authenticate: Basic realm="WRT54G"
Content-Type: text/html
Connection: close
     <HTML><HEAD><TITLE>401 Unauthorized</TITLE></HEAD>
<BODY BGCOLOR="#cc9999"><H4>401 Unauthorized</H4>
Authorization required.
</BODY></HTML>
_EOF_
```

将 fakelinksys.sh 脚本拷贝到/usr/share/honeyd/scripts/目录下，运行 chmod 755 fakelinksys.sh 命令修改 fakelinksys.sh 为可执行。

输入以下命令运行 honeyd

#honeyd –l /var/log/honeyd –p /etc/honeypot/nmap.prints –x /etc/honeypot/xprobe2.conf –d

这个虚拟无线蜜罐系统让你可能看到攻击者会不断尝试默认口令（linksys/admin）。

虚拟无线蜜罐的局限性与传统蜜一样存在隐蔽性的问题，入侵者的水平越来越高，黑客能根据你的系统中的蛛丝马迹分辨出你部署的蜜罐，道高一尺魔高一丈较量使得蜜罐技术不断向前发展[35]。

3.4 本章小结

本章主要深入剖析入侵检测系统 Snort 的模块结构及功能，同时利用 Snort 和 Base（基本安全分析引擎）实现一个协同入侵检测审计系统。通过对无线网络安全状况的实际调查研究，介绍无线安全的认证及加密机制；分析 Honeyd 的运行机理，同时详细阐述利用 Honeyd 部署无线蜜罐。

4 一个主动防御安全系统实现及测试分析

本章主要内容是对蜜网的数据分析，僵尸网络(Botnet)是近年来兴起的危害互联网的重大安全威胁之一，对僵尸网络的发现和跟踪能够帮助安全研究人员深入了解僵尸网络攻击模式网络安全的挑战之一就是你需要了解攻击者，要了解你存在的威胁并保护你自己的资源，你需要了解你的敌人，通过分析统计蜜网的捕获的数据，发现跟踪僵尸网络。通过 Nepenthes 分析了解恶意软件及蠕虫的攻击传播过程。

4.1 蜜网分析跟踪僵尸网络

僵尸网络是近年来兴起的危害互联网的重大安全威胁之一，攻击者通过各种途径传播僵尸程序感染互联网上的大量主机，而被感染的主机将通过一个控制信道接收攻击者的指令，组成一个僵尸网络。大部分的僵尸网络都可以在攻击者的控制下进行进一步的传播，从而使得僵尸网络的规模越来越庞大。一旦攻击者拥有一定规模的僵尸网络，就可以利用僵尸网络所控制的资源，在互联网上建立起了一种强势地位，并且可以利用这些资源获取经济利益。僵尸网络的危害主要体现在发动分布式拒绝服务攻击、发送垃圾邮件以及窃取僵尸主机内的敏感信息等[36]。

Honeynet 可以说是一个学习工具！它是一个专门设计来让人“攻陷”的网络，一旦被入侵者所攻破，入侵者的一切信息、工具等都将被用来分析学习。我们尽我们最大的能力来记录和捕获对 Honeynet 每一个探测，攻击，和使用。这些原始的数据有很高价值。Honeynet 数据有价值的地方是 Honeynet 减少了主动错误信息(false positives)和被动错误信息(false negatives)所产生的问题。主动错误信息(false positives)指的是当组织由于恶意活动而被通知警报时候，经检查其实没有任何事情发生，而当这个组织持续的被主动错误信息(false positives)所触发警报，他们开始忽略他们的警报系统和数据采集，导致警告系统人为的无效[37]。

在我部署的蜜网系统中，使用的拓扑结构为图 2-1，其中蜜网网关(HoneyWall)是整个蜜网系统的数据监控中心，采用桥接方式连接让数据的传输是透明的，避免使用

NAT 方式出现 TTL 的跳数变化而被黑客发觉。数据的分析使用了蜜网项目组的 Walleye 和我组建安全基本分析引擎审计系统（Base）。Walleye 是一个功能强大界面友好的分析系统（如图 4-1），还可以以图表方式生成统计报表，这很大方便了我们分析数据。

由于黑客工具的集成及自动化使得脚本小子只要按动鼠标就可以扫描入侵任何一台存在漏洞的主机，如果系统连接到 Internet 上的时间超过了 24 小时，就很有可能已经被扫描了，如果系统存在安全漏洞，一个星期就有可能已经被攻破了。我们部署的蜜网系统也验证了这一法则。蜜网系统中的一台 windows 2000 蜜罐，使用默认安装方式，加装 sp2 补丁，没有加装任何防火墙或其它安全措施的，部署完毕后接入 Internet 不到一个星期就遭被攻击[38]。

我们的 BASE 安全警告系统和 Honeywall 显示系统被攻破的时间是 2006 年 3 月 1 月。我们的这台 windows2000 蜜罐主机感染了 W32/Rbot-BPT 蠕虫木马（文件大小 258,560 字节，文件名为 nsmscrs.exe）。这是一个 2006 年 1 月 19 日才出现的 IRC 后门特洛伊木马，该蠕虫木马入侵存在以下的漏洞（LSASS (MS04-011), RPC-DCOM (MS04-012), WKS (MS03-049) (CAN-2003-0812), PNP (MS05-039) 及 ASN.1 (MS04-007) 计算机[39]。

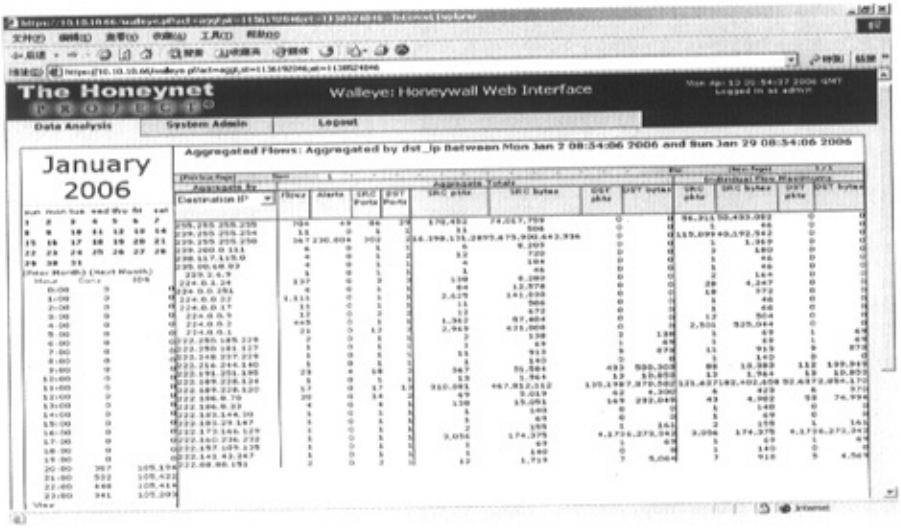


图 4-1 Walleye 系统

4.1.1 BASE 基本安全分析引擎中分析

首先我们查看 BASE，通过查询 2006.2.24 至 2006.3.2 时段内的情况，发现 2006 年 3 月 1 日这天的警报数突增（图 4-2），显然是系统受到的攻击。

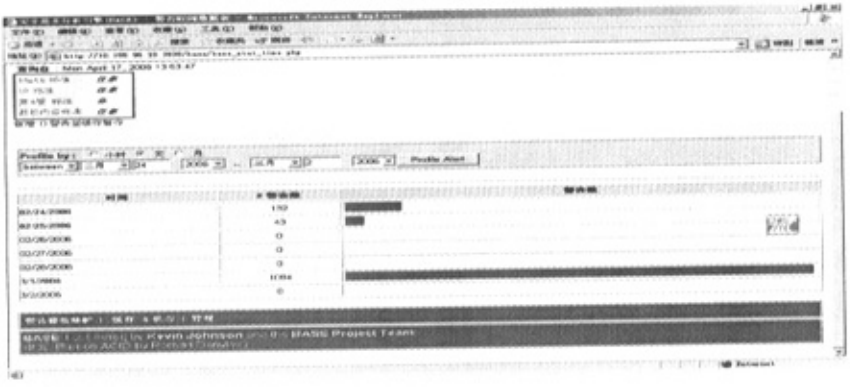
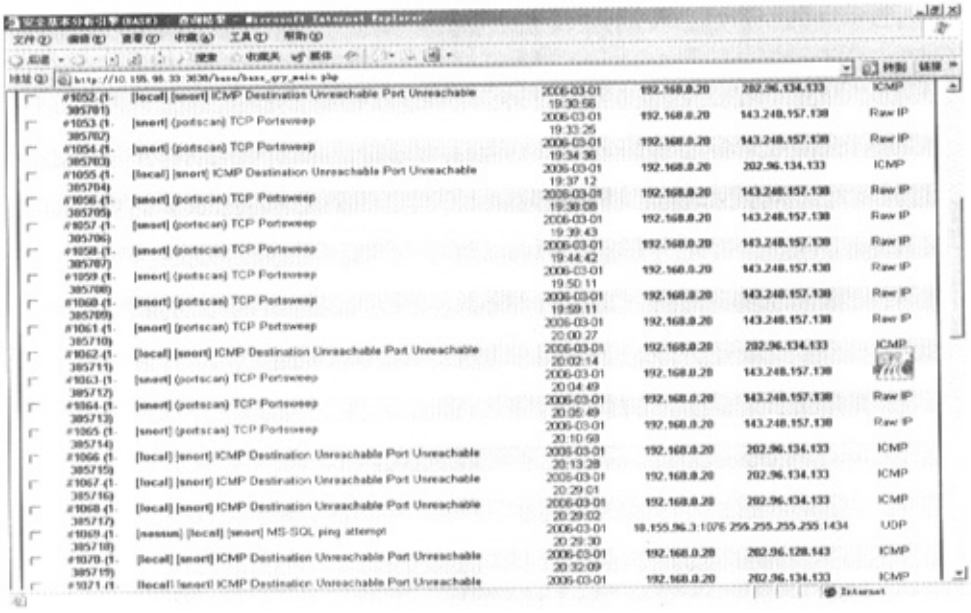


图 4.2 警报数异常情况

在安全基本分析引擎中查看 2006.3.1 详细的警告记录 (图 4-3) ，



#1052 (1-385701)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 19:30:56	192.168.0.20	202.96.134.133	ICMP
#1053 (1-385702)	[snort] (portscan) TCP Portsweep	2006-03-01 19:33:25	192.168.0.20	143.248.157.138	Raw IP
#1054 (1-385703)	[snort] (portscan) TCP Portsweep	2006-03-01 19:34:36	192.168.0.20	143.248.157.138	Raw IP
#1055 (1-385704)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 19:37:12	192.168.0.20	202.96.134.133	ICMP
#1056 (1-385705)	[snort] (portscan) TCP Portsweep	2006-03-01 19:38:08	192.168.0.20	143.248.157.138	Raw IP
#1057 (1-385706)	[snort] (portscan) TCP Portsweep	2006-03-01 19:39:43	192.168.0.20	143.248.157.138	Raw IP
#1058 (1-385707)	[snort] (portscan) TCP Portsweep	2006-03-01 19:44:42	192.168.0.20	143.248.157.138	Raw IP
#1059 (1-385708)	[snort] (portscan) TCP Portsweep	2006-03-01 19:50:11	192.168.0.20	143.248.157.138	Raw IP
#1060 (1-385709)	[snort] (portscan) TCP Portsweep	2006-03-01 19:59:11	192.168.0.20	143.248.157.138	Raw IP
#1061 (1-385710)	[snort] (portscan) TCP Portsweep	2006-03-01 20:00:27	192.168.0.20	143.248.157.138	Raw IP
#1062 (1-385711)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 20:02:14	192.168.0.20	202.96.134.133	ICMP
#1063 (1-385712)	[snort] (portscan) TCP Portsweep	2006-03-01 20:04:49	192.168.0.20	143.248.157.138	Raw IP
#1064 (1-385713)	[snort] (portscan) TCP Portsweep	2006-03-01 20:05:49	192.168.0.20	143.248.157.138	Raw IP
#1065 (1-385714)	[snort] (portscan) TCP Portsweep	2006-03-01 20:10:58	192.168.0.20	143.248.157.138	Raw IP
#1066 (1-385715)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 20:13:28	192.168.0.20	202.96.134.133	ICMP
#1067 (1-385716)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 20:29:01	192.168.0.20	202.96.134.133	ICMP
#1068 (1-385717)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 20:29:02	192.168.0.20	202.96.134.133	ICMP
#1069 (1-385718)	[nessus] [local] [snort] MS-SQL ping attempt	2006-03-01 20:29:30	10.155.96.3:1076	255.255.255.255:1434	UDP
#1070 (1-385719)	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01 20:32:09	192.168.0.20	202.96.128.143	ICMP
#1071 (1-	[local] [snort] ICMP Destination Unreachable Port Unreachable	2006-03-01	192.168.0.20	202.96.134.133	ICMP

图 4-3 BASE 2006 年 3 月 1 日详细警告

#1052-(1-385701) [local] [snort] ICMP Destination Unreachable Port Unreachable 2006-03-01 19:30:56 192.168.0.20 202.96.134.133 ICMP

#1053-(1-385702) [snort] (portscan) TCP Portsweep 2006-03-01 19:33:25 192.168.0.20 143.248.157.138 Raw IP

#1054-(1-385703) [snort] (portscan) TCP Portsweep 2006-03-01 19:34:36 192.168.0.20 143.248.157.138 Raw IP

#1055-(1-385704) [local] [snort] ICMP Destination Unreachable Port Unreachable 2006-03-01 19:37:12 192.168.0.20 202.96.134.133 ICMP

#1056-(1-385705) [snort] (portscan) TCP Portsweep 2006-03-01 19:38:08 192.168.0.20 143.248.157.138 Raw IP

从#1052-(1-385701)警告中看到蜜罐的 IP 地址 192.168.0.20 不停地请求 DNS 202.96.134.133 解释而被我们蜜网网关中的 IPTABLES 所阻挡，入侵的时间是 2006-3-01，从警告的记录中看到 RAW IP 是 143.248.157.138,通过 whois 查询到该主机是 Hostnames : fft1.kaist.ac.kr，这是来自韩国的一台主机。由于警告记录 #1053-(1-385702)看到入侵者启用了通过客户端主机 24.160.180.3:5226 远程控制 143.248.157.138:6663。图 4-5 显示 Base 详细警告记录情况。

图 4-5 Base 详细警告记录图

长度 = 148

Priority Count: 5

Connection Count: 19

IP Count: 15

Scanned IP Range: 24.160.180.3:143.248.157.138

Port/Proto Count: 15

Port/Proto Range: 5226:6663

以上是安全基本分析引擎中的 playload，6663 是 IRC 服务器端使用的端口，因此 143.248.157.138 是一台被控的 IRC 服务器，这表明我们的蜜罐已经变成了黑客的一台僵尸网络。

4.1.2 蜜网网关 honeywall 跟踪僵尸网络

前面我们通过基本安全分析引擎了解到蜜罐感了 W32/Rbot-BPT 蠕虫木马,下面我们通过分析蜜网网关上的数据来跟踪感染木马的蜜罐。Honeywall 所有的日志都记录在/var/log 目录下，主要包括入侵检测 Snort、数据控制 snort_inline、被动系统指纹 p0f、防火墙 iptables 等日志[41]。

先来分析 snort/2006301 中的 summary.log，这是 Snort 日志的概要统计：

Top 10 Remote IPs:

Remote IP	Packets	Bytes	Conns
192.168.0.20	8015	263264	2041
143.248.157.138	2740	0	692
202.96.134.133	1677	104005	526
24.160.180.3	1410	0	460
202.96.128.143	652	40284	314
10.155.185.84	604	0	302
10.155.234.3	587	0	294

10.155.12.9	561	0	282
10.155.71.45	344	0	173
10.155.92.80	334	0	168

以上的数据表明木马感染蜜罐主机后开始扫描我们局域网的其它网段，发现并感染存在漏洞的主机。由 Snort 日志得到了下面的信号特征数据：

[**] [111:2:1] (spp_stream4) possible EVASIVE RST detection [**]
03/01-17:12:03.209250 194.109.11.65:6556 -> 192.168.0.20:1607
TCP TTL:40 TOS:0x0 ID:17444 IpLen:20 DgmLen:40 DF
*****R** Seq: 0xAA098C6B Ack: 0x0 Win: 0x0 TcpLen: 20

主机 194.109.11.65 正在以 6556 端口连接蜜罐。

P0f 中的日志检测到蜜罐 IP192.168.0.20 正在连接 IRC 服务器 143.248.157.138 和控制端主机 24.160.180.3，远端连接端口分别使用 6663 及 5226。

<Wed Mar 1 21:16:01 2006> 192.168.0.20:4789 - Windows 2000 SP2+, XP SP1 (seldom 98 4.10.2222) -> 143.248.157.138:6663 (distance 0, link: ethernet/modem)
<Wed Mar 1 21:16:01 2006> 192.168.0.20:4788 - Windows 2000 SP2+, XP SP1 (seldom 98 4.10.2222) -> 24.160.180.3:5226 (distance 0, link: ethernet/modem)

以下是 Iptables 的日志，表明 IPTABLES 拦截了三个要进来的数据包，源地址为 192.168.0.20,目标地址分别是 24.160.180.3:5226、143.248.157.138:6663、68.117.123.25:5226。由此可知虽然我们的蜜罐被木马控制着，但黑客无法利用我们的蜜罐作跳板入侵其它的系统，蜜网系统中所有的数据都是可监视、可控制的。

Mar 1 17:04:16 roo-test kernel: INBOUND TCP: IN=br0 OUT=br0 PHYSIN=eth0 PHYSOUT=eth1 SRC=192.168.0.20 DST=24.160.180.3 LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=32793 DF PROTO=TCP SPT=4587 DPT=5226 WINDOW=16384 RES=0x00 SYN URGP=0

Mar 1 17:04:16 roo-test kernel: INBOUND TCP: IN=br0 OUT=br0 PHYSIN=eth0 PHYSOUT=eth1 SRC=192.168.0.20 DST=143.248.157.138 LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=32794 DF PROTO=TCP SPT=4585 DPT=6663 WINDOW=16384 RES=0x00 SYN URGP=0

Mar 1 17:04:22 roo-test kernel: INBOUND TCP: IN=br0 OUT=br0 PHYSIN=eth0 PHYSOUT=eth1 SRC=192.168.0.20 DST=68.117.123.25 LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=33206 DF PROTO=TCP SPT=4589 DPT=5226 WINDOW=16384 RES=0x00 SYN URGP=0

4.1.3 僵尸网络的防范

罪大恶极的黑客正在劫持和连接数千台被攻破的计算机组成自动程序网络（botnets）发动协调一致的分布式拒绝攻击。一个攻击者可能利用数千台大企业的主机实施攻击。还有很多黑客使用好几套计算机实施攻击。防御 botnets 攻击是非常困难和复杂的。而且僵尸网络控制中心节点启遍布世界各地（图 4-6）[42],使得防范更显得团难重重，但是我们通过以下五个实用步骤可以识别和消除网络中被 bot 感染的主机。

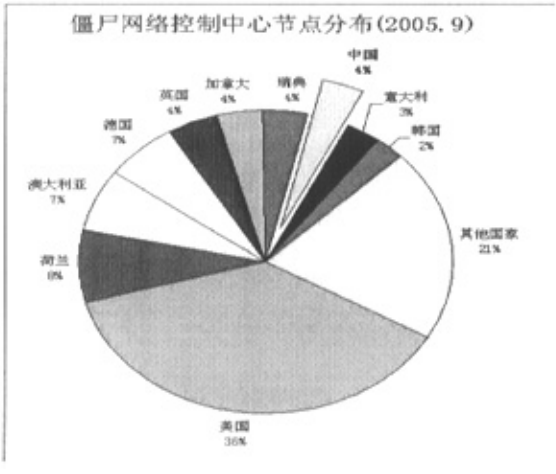


图 4-6 僵尸网络控制中心节点分布图

（1） 扫描主机和网络通信查找 bot 感染泄漏的痕迹。

（2） 调节路由器、防火墙和入侵检测/防御系统，以监测和封锁 botnet 通信。

（3） 加密主机，防止 botnet 利用漏洞。

（4） 应用逆向工程对付 bot 代码。

（5） 与执法部门合作起诉 botnet 的制造者。

4.2 Nepenthes 对抗恶意软件及蠕虫

恶意软件、网络蠕虫及垃圾邮件是当今网络安全中最令人头疼问题，连微软的比尔盖茨也拿恶意软件没办法。第一个的网络蠕虫叫"莫里斯蠕虫"，它诞生于 1988 年。每一种新网络蠕虫发作造成的经济损失难以估算，2001 年爆发的"尼姆达"病毒曾造成了企业 635 亿美元的损失，大家记忆犹新的冲击波(Blaster)破坏力更强[43]。

Nepenthes 源于恶意软件收集器 mwcollect，都是 UNIX 下基于 GPL 的开源软件。Nepenthes 是一个模拟不同弱点漏洞达到收集恶意软件的低交互蜜罐。部署在蜜网系中可以收集统计各种知名漏洞的攻击，快速帮助我们找出网络中哪些机器中了病毒并生成预警报表。

4.2.1 Nepenthes 体系结构

Nepenthes 其体系统结构图（如图 4-7），主要由以下几部分组成：

- 模组构架
  - 弱点漏洞模块
  - 脚本代码处理
  - 下载模块
  - 提交模块
- 事件触发器
- 脚本模拟
- 模拟主动攻击漏洞服务，具体的知名漏洞参见表 4.1[44]。

表 4.1 知名漏洞列表

漏洞模块名称	知名漏洞名称	端口
Vuln_dcom	Dcom 漏洞 ms03-039	135、137、138、445、593
Vuln_asn1	ASN.I ms04-007	445
Vuln_lsass	LSASS 漏洞 ms04-011	135、137、138、445、593

续表 4.1

Vuln_wins	WINS 漏洞 ms04-045	42、80
Vuln_netdde	MS04-031	139
Vuln_iis	Unicode	80
Vuln_upnp	MS05-039	445、10000
Vuln_msmq	MS05-017	135、137、138、445、593、1801、2101、2103、3527
Vuln_netbiosname		TCP/443
Vuln_mssql	MSSQL Hello Buffer Overflow 等	TCP/1433、UDP1434
Vuln_sasserftpd	LSASS 漏洞 MS04-011	135、139、445、1023、1025、5554、

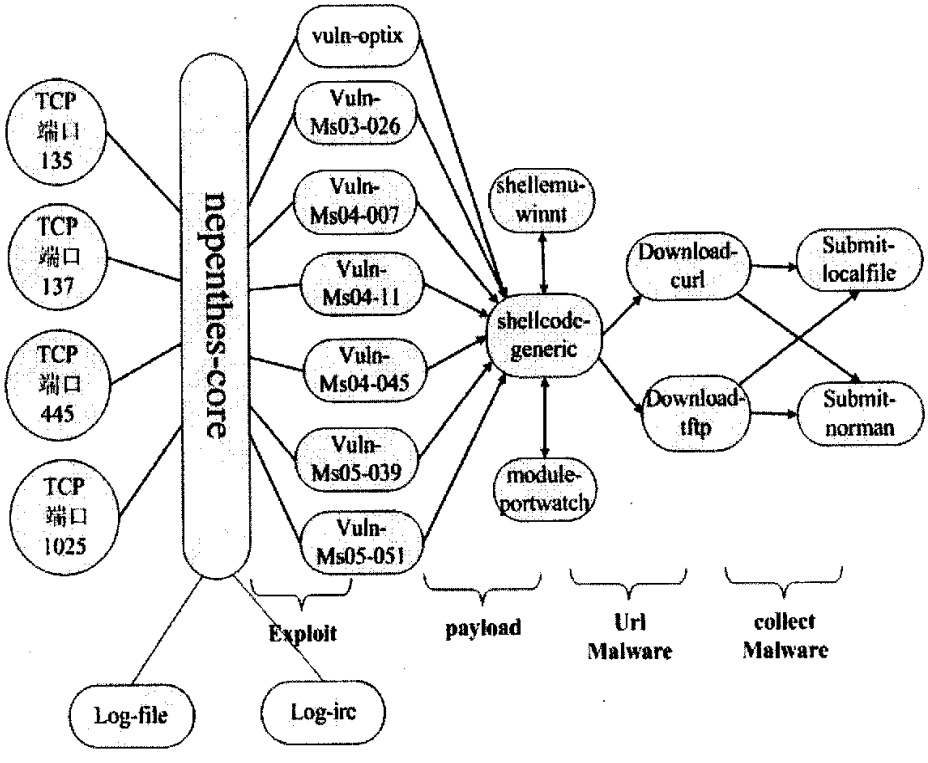


图 4-7 nepenthes 体系结构图

Nepenthes 根据知名的漏洞模块在不同的端口上模拟出不同漏洞服务，然后等待蠕虫的攻击，比如模拟 Vuln_sasserftpd 的漏洞，下面的模拟开放 ftp 服务端口 5554 和 1023。

[02042006 13:24:06 spam net mgr] bindTCPSocket 0 5554 0 30 8090cd0

[02042006 13:24:06 spam net handler] <in virtual bool nepenthes::TCPSocket::bindPort()>

[02042006 13:24:06 debug net handler] Success binding Port 5554

[02042006 13:24:06 debug net] Socket TCP (bind) 0.0.0.0:0 -> 0.0.0.0:5554

Adding DialogueFactory SasserFTPD Factory

[02042006 13:24:06 spam net mgr] bindTCPSocket 0 1023 0 30 8090cd0

[02042006 13:24:06 debug net handler] Success binding Port 1023

[02042006 13:24:06 debug net] Socket TCP (bind) 0.0.0.0:0 -> 0.0.0.0:1023

Adding DialogueFactory SasserFTPD Factory

将知名的漏洞端口通过 socket 函数绑定处于监听状态，然后通过 DialogueFactory() 创建一个对应的漏洞会话并调用相应模块实现漏洞虚拟服务，当远程入侵主机触发某个事件时，nepenthes 会根据触发事件的条件调用相应脚本代码处理来对入侵的蠕虫进行响应[45]。

为了能更清楚了解 nepenthes 的工作机制，图 4-8 给出 nepenthes 的类图模型。

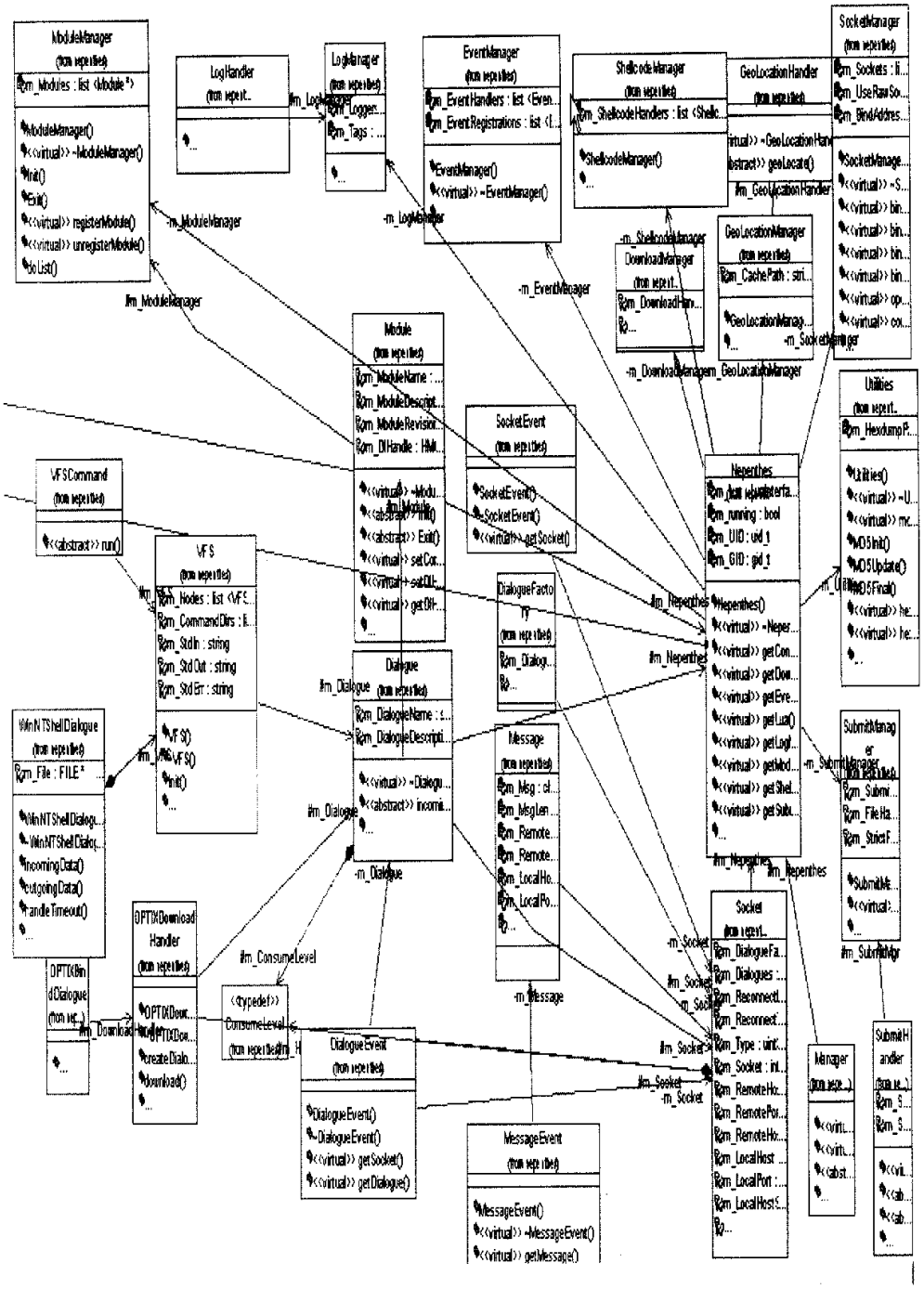


图 4-8　nepenthes 的类图模型

4.2.2 nepenthes 分析蠕虫攻击传播过程

1）恶意软件收集工具 nepenthes 部署的工作环境：

使用虚拟主机软件 Vmware5.5

主机:Windows 2003　512M 内存 CPU 2.4 G

客户机: 运行 2.6.9-kanotix-8 内核的 Knoppix Linux 发行版

运行后 nepenthes，由 logged_downloa 日志文件统计一共捕获到 4 种文件名的恶意软件(如表 4.2)

表 4.2　收集恶意软件文件名统计

恶意软件文件名	次数
winxSysten32.exe	21
sistem32.exe	284
eraseme_5 位随机生成数字.exe	68
setup_5 位随机生成数字.exe	3

root@auditor:/var/log/nepenthes# cat logged_downloads

[2006-04-10T15:56:43] ftp://1:1@10.155.252.33:32910/winxSysten32.exe

[2006-04-10T15:56:44] ftp://1:1@10.155.252.33:32910/winxSysten32.exe

[2006-04-10T05:41] ftp://1:1@10.155.97.176:1978/setup_46458.exe

06-04-16T16:09:34] ftp://1:1@10.155.248.155:33644/Sistem32.exe

[2006-04-16T16:13:10] ftp://1:1@10.155.195.199:21800/Sistem32.exe

[2006-04-16T16:16:17]ftp://1:1@10.155.212.203:31843/Sistem32.exe

[2006-04-14T04:57:10] ftp://1:1@10.155.97.77:3721/eraseme_58152.exe

由日志可以看出，虽然蠕虫下载使用的文件名各不相同，但都是想通过使用户名密码都是 1 将蠕虫 ftp 下载感染的主机中。

2）nepenthes 分析恶意软件攻击传播详细过程

我们通过分析 nepenthes.log 日志文件可以清楚了解蠕虫是如何攻击传播详细过程。

（1）蠕虫宿主 10.155.248.16 发现并攻击 nepenthes 模拟存在知名漏洞 ASN1 主机 10.155.99.234，nepenthes 的事件触发器启动。

[02042006 13:29:15 spam mgr event] <in virtual uint32_t nepenthes::EventManager::handleEvent(nepenthes::Event*)>

[02042006 13:29:19 debug net mgr] Socket TCP (bind) 0.0.0.0:0 -> 0.0.0.0:80

DialogueFactory ASN1 Dialogue Factory creates dialogues for the SMB and IIS flaw killbill showed us could Accept a Connection

[02042006 13:29:19 spam net handler] <in virtual nepenthes::Socket* nepenthes::TCPSocket::acceptConnection()>

[02042006 13:29:19 spam net handler] Socket TCP (accept) 10.155.248.16:4126 -> 10.155.99.234:80

[02042006 13:29:19 spam net handler] Adding Dialogue ASN1 Dialogue Factory

（2）然后以不同字节对 SMB and IIS 进行 Overflow 溢出入侵攻击

[02042006 13:29:20 spam sc handler] Shellcode is 1460 bytes long

[02042006 13:29:20 info sc handler] Found ASN1Base64 .. 1460

[02042006 13:29:20 debug sc mgr] SCHMGR REPROCESS Msg ptr is 807da10

.....

[02042006 13:29:20 spam sc handler] Shellcode is 1050 bytes long

.....

[02042006 13:29:20 spam sc handler] Shellcode is 3508 bytes long

[02042006 13:29:20 info sc handler] Found ASN1Base64 .. 3508

[02042006 13:29:20 debug sc mgr] SCHMGR REPROCESS Msg ptr is 807da10

（3）溢出攻击成功后蠕虫利用 echo 出成一个名为“o”的文件并执行 ftp 下载命令

[02042006 13:29:20 info sc handler] Detected generic CMD Shellcode: "cmd /k echo open 10.155.248.16 27134 > o&echo user 1 1 >> o &echo get Sistem32.exe >> o &echo quit >> o &ftp -n -s:o &del /F /Q o &Sistem32.exe"

由上可以看到文件“o”的内容为：

Open 10.155.248.16 27134

User 1 1

Get Sistem32.exe

Quit

接着蠕虫执行以下 ftp 命令传输下载蠕虫感染文件 Sistem32.exe

ftp –n –s：o

-n 参数是 ftp 自动登录连接，-s：o 参数是指出自动执行刚才建立的文件 o 中所有命令。

下载完蠕虫病毒文件 Sistem32.exe,运行 del /F /Q o 命令使用安静模式强制删除 FTP 文本命令集文件 o,然后运行蠕虫病毒文件 Sistem32.exe。如图 4-9

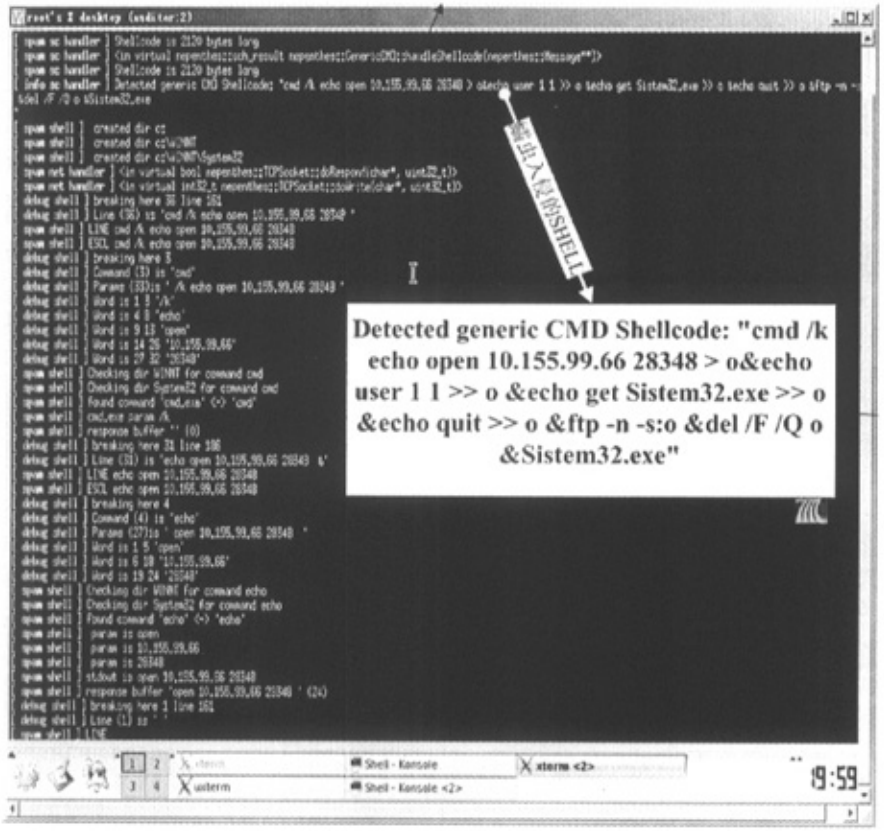


图 4-9 漏洞蠕虫 FTP 传播命令图

（4）nepenthes 回应蠕虫攻击的脚本处理

[02042006 13:29:20 spam shell] created dir c:

[02042006 13:29:20 spam shell] created dir c:\WINNT

[02042006 13:29:20 spam shell] created dir c:\WINNT\System32

[02042006 13:29:20 spam net handler] <in virtual bool nepenthes::TCPSocket::doRespond(char*, uint32_t)>

[02042006 13:29:20 spam net handler] <in virtual int32_t nepenthes::TCPSocket::doWrite(char*, uint32_t)>

[02042006 13:29:20 debug shell] breaking here 37 line 161

[02042006 13:29:20 debug shell] Line (37) is 'cmd /k echo open 10.155.248.16 27134

[02042006 13:29:20 spam shell] LINE cmd /k echo open 10.155.248.16 27134

……

每个蠕虫都有自己的特征，如图 4-10 是我们捕抓到的蠕虫传播文件名及其它一些特征值[46]。

图 4-10 蠕虫文件名及其 hash 值

从图 4-10 可知我们我们分析的 sistem32.exe 蠕虫属于 W32/Spybot.gen3 类的恶意间谍木马。除了这个外我们还捕获到 eraseme_5 位随机生成数字.exe 类型的未知蠕虫，其入侵传播的方式与 sistem32.exe 蠕虫大致相同，不同的是其利用 MS05-039 漏洞攻

击 445 端口。

3）蠕虫传播模式及防范策略

通过以上的分析，我们对蠕虫的结构及传播过程有了很深的认识[47]。

（1）蠕虫的基本程序结构为：

①传播模块：负责蠕虫的传播，这是本文要讨论的部分。

②隐藏模块：侵入主机后，隐藏蠕虫程序，防止被用户发现。

③目的功能模块：实现对计算机的控制、监视或破坏等功能。

传播模块由可以分为三个基本模块：扫描模块、攻击模块和复制模块。

（2）蠕虫程序的一般传播过程为：

①扫描：由蠕虫的扫描功能模块负责探测存在漏洞的主机。当程序向某个主机发送探测漏洞的信息并收到成功的反馈信息后，就得到一个可传播的对象。

扫描过程是随机选取某一段 IP 地址，然后对这一地址段上的主机扫描。因为时刻扫描网络，所以会发送大量的数据包，造成网络拥塞，影响网络通信速度。

扫描发送的探测包是根据不同的漏洞进行设计的。比如，针对远程缓冲区溢出漏洞可以发送溢出代码来探测，针对 web 的 cgi 漏洞就需要发送一个特殊的 http 请求来探测。当然发送探测代码之前首先要确定相应端口是否开放，这样可以提高扫描效率。一旦确认漏洞存在后就可以进行相应的攻击步骤，不同的漏洞有不同的攻击手法，只要明白了漏洞的利用方法，在程序中实现这一过程就可以了。前面 nepenthes 实际数据分析让我们很清楚看到，蠕虫是如何利用漏洞这一过程。

②攻击：攻击模块按漏洞攻击步骤自动攻击步骤 1 中找到的对象，取得该主机的权限（一般为管理员权限），获得一个 shell。

③复制：复制模块通过原主机和新主机的交互将蠕虫程序复制到新主机并启动。

攻击成功后，一般是获得一个远程主机的 shell，对 win2k 系统来说就是 cmd.exe，得到这个 shell 后我们就拥有了对整个系统的控制权。复制过程也有很多种方法，可以利用系统本身的程序实现，也可以用蠕虫自代的程序实现。复制过程实际上就是一个文件传输的过程，实现网络文件传输很简单。一般是利用 ftp 或 tftp 等传输模式。

除了以上一般的传播模式之外，邮件进行自动传播甚至 html 页面也是最常见传播

模式。

④如何从安全防御的角度看蠕虫的传播模式

我们通过剖析蠕虫传播模式及传播过程后,防范蠕虫的安全策略可以从下几方面进行。对蠕虫传播的一般模式来说，我们目前做的安全防护工作主要是针对其第二环即"攻击"部分，为了防止攻击，要采取的措施就是及早发现漏洞并打上补丁。其实更重要的是第一环节的防护，对扫描的防护现在人们常用的方法是使用防护墙来过滤扫描。使用防火墙的方法有局限性，因为很多用户并不知道如何使用防火墙，所以当蠕虫仍然能传播开来，有防火墙保护的主机只能保证自己的安全，但是网络已经被破坏了。另外一种方案是从网络整体来考虑如何防止蠕虫的传播，就是可以通过蜜网系统捕获到蠕虫的入侵特征串，以便更好针对这些特征串制定入侵检测规则。

构建主动防御的蜜网可以让我们及时、深入的了解未知的入侵和攻击，更好对抗恶意软件。

4.4 本章小结

在本章主要通过利安全基本分析引擎和蜜网网关 HoneyWall 进行数据分析，通过分析发现跟踪僵尸网络的活动，同时给出了对其防范几点措施。利用 Nepenthes 收集恶意软件及蠕虫的信息及对深入剖析。

5 结束语

5.1 工作总结

攻击与防御永远是一对不可调和的矛盾。本文对主动防御安全系统的关键技术蜜网系统进行了研究，重点设计并且实现了基于主动安全策略的蜜网系统及对其进行深度的研究分析。主要完成的工作如下：

（1）主要就蜜罐定义及蜜网系统体系结构，详细阐述了蜜网系统的三大核心技术：数据捕获、数据控制、数据分析。给出实现蜜网体系结构及实现方法。

（2）研究基于内核层 sebek 调用解决数据捕获中的如何捕获入侵者加密信息。

（3）深入研究入侵检测系统 snort 的运行体系结构模型及日志数据库关系，并结合 base 实现安全基本分析引擎审计系统，本安全基本分析引擎系统是分布式协同入侵检测的控制中心，基于友好中文管理界面，实现实时入侵警告，分类查询生成各种中文图形报表。

（4）调查研究当前无线网络安全状况及介绍无线网络的认证加密算法，同时利用 Honeyd 部署一个虚拟无线蜜罐系统。

（5）通过深入分析部署蜜网系统数据发现跟踪僵尸网络，利用 nepenthes 剖析恶意软件及网络蠕虫攻击传播的详细过程。

5.2 未来研究展望

由于对网络安全的认识及知识水平有限的原因，对基于蜜罐和蜜网的主动防御安全系统研究还存在不足之处，构建的蜜网系统有还待完善和改进。

下一阶段工作要点有：

（1）继续深入研究蜜网组成原理细节。

（2）扩大现在蜜网系统的规模,组建小型的蜜场。

（3）在现在的基础之上继续对安全基本分析引擎的开发和研究,以便对蜜网系统的数据分析提供更完善分析报告。

的数据分析提供更完善分析报告。

（4）研究基于 Windows 平台的无线网卡监视模式的程序开发。研究国家自主知识产权的无线局域网 WAPI 标准，提出一些有建设性的方案。

（5）进一步研究 Nepenthes 系统与 Iptables 的联动，提高其发现未知恶意软件的智能，改进使其成为收集、警报、防范恶意软件于一身的主动防御系统。

（6）研究利用蜜网技术进行分析取证。

附录 A　改进 BASE 对中文支持源程序

（1）、使用如下 php 代码生成中文汉字 GBK 码表

```
<?
//生成 gbk.txt
$st="";
for ($i=1; $i<88; $i++){
for ($j=1; $j<95; $j++){
$st.=chr($i+160).chr($j+160)."|";
}
}
echo $st;
$fn=fopen("gbk.txt","w");
fputs($fn,$st);
fclose($fn);
?>
```

用写字板打开 GBK.txt,别存为 UTF8 格式文件的 UTF8.txt。将这两个文件放到 base 目录下。

（1）、修改图形报表模块程序

在 base_graph_common.php 中加入如下代码

```
function gb2utf8($string)
{
$ut="";
//读出两个编码的字符集合文本
$gb=file("gbk.txt");
$gb=explode("|",$gb[0]);
$utf=file("utf8.txt");
$utf=explode("|",$utf[0]);
//为了方便查询，把$gb 数组的 key 与 values 交换；
```

```
$gb=array_flip($gb);
while ($string){
$st=substr($string,0,1);
$string=substr($string,1);
if (ord($st)<128){
//单字节直接送回
$ut.=$st;
} else {
//双字节的处理
$st.=substr($string,0,1);
$string=substr($string,1);
$ut.=$utf[$gb[$st]];
}
}
return $ut;
}
```

修改 base_graph_display.php

```
//增加对中文字体的支持 support simplechinse --linuxfly
  $gbtitle=gb2utf8($title);//通过 gb2utf8（见 base_graph_common.php）将中文转换成 utf8 格式
  //定义汉字字体，否则会产生乱码
  $Font =& $Graph->addNew('font', 'simfang'); //simfang 为 ttf 中文字体
  $Font->setSize(12);
  $Graph->setFont($Font);
```