# 蜜罐技术及其应用

诸葛建伟
北京大学狩猎女神项目组
The Artemis Project

北京大学计算机科学技术研究所

# 狩猎女神项目组介绍

- 狩猎女神项目组(The Artemis Project)
  - 北京大学计算机研究所信息安全工程研究中心
  - 世界蜜网研究联盟中国唯一团队
  - 项目网站: www.honeynet.org.cn
  - HoneynetCN邮件组 http://groups.yahoo.com/group/honeynetcn/
- 项目组重要事件
  - 2004.9 成立
  - 2005.2 加入世界蜜网研究联盟
  - 2005.11.11 国家项目“基于蜜网技术的僵尸网络监测系统”启动
  - 2005.12 协助中国移动、CNCERT/CC完成信产部安全应急演练
  - 2005.12.15 截获黛蛇蠕虫，并协助CNCERT/CC抑制进一步传播，准确定位了蠕虫编写者
  - 2006.7 协助CNCERT/CC在全国范围部署分布式蜜网
  - 2006.8 受邀参加世界蜜网研究联盟年会

# 讲师介绍

- 诸葛建伟博士
  - 1997-2006就读北京大学，获理学博士学位
  - 北京大学计算机科学技术研究所助理研究员
  - 狩猎女神项目组发起人及技术负责人
  - 研究方向：蜜罐与蜜网技术，入侵检测/关联分析，网络攻防知识建模，恶意软件分析及防范技术
  - 第一作者发表国际会议、核心期刊论文十余篇，申请发明专利4项
  - 2004年微软学者，2005年IBM全球博士生英才，2005年北大五四青年科学奖一等奖
  - Email: zhugejianwei@icst.pku.edu.cn

# “蜜罐技术及其应用”课程安排

- 第一讲：蜜罐技术
  - 演示—典型蜜罐工具
- 第二讲：蜜网技术
  - 演示—虚拟蜜网
- 第三讲：蜜罐技术发展趋势
- 第四讲：蜜罐技术的应用
  - 演示—基于蜜网技术的僵尸网络监测系统

# 提问规则

- 课程中间休息10分钟
  - 仅限对讲解内容相关的问题
  - 对问题不感兴趣的可自由活动
- 课程结束后留充分时间问答
  - 欢迎任何问题
  - zhugejianwei@icst.pku.edu.cn

# 第一讲：蜜罐技术

诸葛建伟
北京大学狩猎女神项目组
The Artemis Project

# 蜜罐技术

- 蜜罐技术的提出与发展历程
- 蜜罐技术的概念和原理
- 蜜罐技术的分类
- 典型蜜罐工具
  - Honeyd (*nix平台)
  - KFSensor (Win32平台)

# 蜜罐技术

- 蜜罐技术的提出与发展历程
- 蜜罐技术的概念和原理
- 蜜罐技术的分类
- 典型蜜罐工具
  - Honeyd (*nix平台)
  - KFSensor (Win32平台)

# 互联网安全状况

- 安全基础薄弱
  - 操作系统/软件存在大量漏洞
  - 安全意识弱、缺乏安全技术能力
- 任何主机都是攻击目标！
  - DDoS、跳板攻击需要大量僵尸主机
  - 蠕虫、病毒的泛滥
  - 并不再仅仅为了炫耀：Spamming, Phishing
- 攻击者不需要太多技术
  - 攻击工具的不断完善
    - Metasploit: 40+ Exploits
  - 攻击脚本和工具可以很容易得到和使用
    - 0-day exploits: packetstorm

# 网络攻防的非对称博弈

- 工作量不对称
  - 攻击方：夜深人静，攻其弱点
  - 防守方：24*7，全面防护
- 信息不对称
  - 攻击方：通过网络扫描、探测、踩点对攻击目标全面了解
  - 防守方：对攻击方一无所知
- 后果不对称
  - 攻击方：任务失败，极少受到损失
  - 防守方：安全策略被破坏，利益受损
- 攻击方掌握主动权

# 传统安全防护机制的不足

- 被动安全防护机制
  - 加密、VPN
  - 防火墙：配置问题、针对开放业务端口的攻击、内部攻击
  - 入侵检测系统IDS：已知攻击特征库、高误报率
  - 反病毒软件：病毒特征库在线升级，延迟
- “主动”安全防护机制
  - 漏洞扫描与补丁分发工具:扫描脚本、补丁延迟
  - 入侵防御系统IPS：已知攻击特征库、“傻瓜式”

# 蜜罐技术的提出

- 防御方尝试改变攻防博弈不对称性而提出的一种主动防护技术
  - 对攻击者的欺骗技术—增加攻击代价、减少对实际系统的安全威胁
  - 了解攻击者所使用的攻击工具和攻击方法
  - 追踪攻击源、攻击行为审计取证
- 蜜罐技术的提出
  - Honeypot：首次出现在Cliff Stoll的小说“The Cuckoo’s Egg”(1990)
  - 著名计算机安全专家Fred Cohen
    - 第一个开源蜜罐工具DTK：Deception Tool Kit (1997)
    - 蜜罐技术的理论基础：A Framework for Deception (2001)

# 蜜罐技术的发展历程

- 蜜罐技术
  - 1998年后，出现DTK、Honeyd等大量开源蜜罐工具
  - 同期出现一些商业产品，但并未得到市场普及
- 蜜网技术
  - 1999年由蜜网项目组(The Honeynet Project)提出并实现
  - 目前已发展到第三代蜜网技术
- 蜜场技术
  - 2003年由Lance Spitzner首次提出Honeypot farms思想
  - 目前仍未有实际的工具、产品和应用

# 蜜罐技术

- 蜜罐技术的提出与发展历程
- 蜜罐技术的概念和原理
- 蜜罐技术的分类
- 典型蜜罐工具
  - Honeyd (*nix平台)
  - KFSensor (Win32平台)

# 蜜罐技术的概念

- 蜜罐的定义
  - “A security resource who’s value lies in **being probed, attacked or compromised**”
  - 一类安全资源，其价值就在于被探测、被攻击及被攻陷
- 蜜罐技术原理—“蜜罐公理”
  - 无任何业务用途→没有任何的正常活动→任何活动都是恶意的
  - 攻击诱骗、安全威胁预警
  - 绕过攻击检测问题—区分正常业务和攻击行为
    - 防火墙：定义安全策略保证正常业务
    - 入侵检测系统：根据已知攻击特征进行检测
    - 反病毒软件：根据已知病毒特征码

# 蜜罐技术—如何实施诱骗？

- 欺骗环境(Pot)的构建：黑洞 VS. 模拟 VS. 真实
  - 零交互式蜜罐：黑洞，没有任何响应
  - 低交互式蜜罐—虚拟蜜罐：模拟网络拓扑、协议栈、服务(Honeyd/Nepenthes)；模拟OS (Sandbox)
  - 高交互式蜜罐
    - 物理蜜罐：完全真实的硬件、OS、应用、服务
    - 虚拟机蜜罐：模拟的硬件(VMWare)/真实的OS、应用、服务
- 部署陷阱，诱骗攻击者(Honey)
  - 守株待兔：安全漏洞—针对扫描式攻击
  - 酒香也怕巷子深：散播陷阱信息，引诱攻击者 (Google Hacking Honeypot, HoneyEmail)
  - 重定向技术 (Honeyfarm)
  - 主动出击：利用爬虫技术—客户端蜜罐(HoneyClawer 恶意网站监测)

# 蜜罐技术—诱骗之后

- 欺骗环境的核心功能需求
  - 数据控制
  - 数据捕获
  - 数据分析
  - 欺骗环境的配置管理
- 欺骗与反欺骗的较量
  - 欺骗环境伪装：环境伪装/业务伪装
  - 对欺骗环境的识别：fingerprinting
  - Anti-Honeypot, Anti-Anti-Honeypot, ... — 更深一层的博弈问题


# 蜜罐技术优势

- 基于蜜罐技术的原理，蜜罐技术具有以下优势：
  - 具有高度分析价值的小数量集
  - 极低（近乎为0）的误报率
  - 对未知攻击的发现能力
  - 对加密通讯、IPv6等环境具有同样的适应性
  - 资源需求低，投入/产出比大
  - 原理简单，“简单往往意味着实用”

# 蜜罐技术的弱势

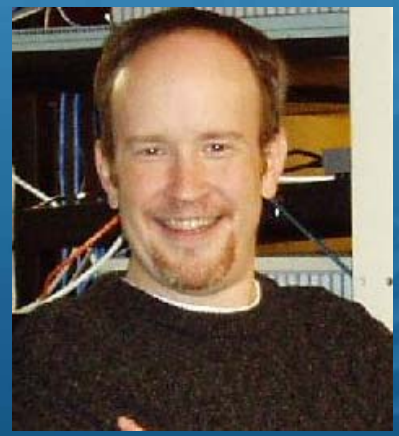
- 传统蜜罐技术的弱势
  - 监控视野较小：只能发现蜜罐中的攻击行为
  - 欺骗能力较低：容易被高水平黑客所识别
  - 可能带来安全风险
- 如何克服这些弱势？
  - Let us think about it.
  - 在第三讲《蜜罐技术发展趋势》中进行讨论

# 蜜罐技术

- 蜜罐技术的提出与发展历程
- 蜜罐技术的概念和原理
- 蜜罐技术的分类
- 典型蜜罐工具
  - Honeyd (*nix平台)
  - KFSensor (Win32平台)

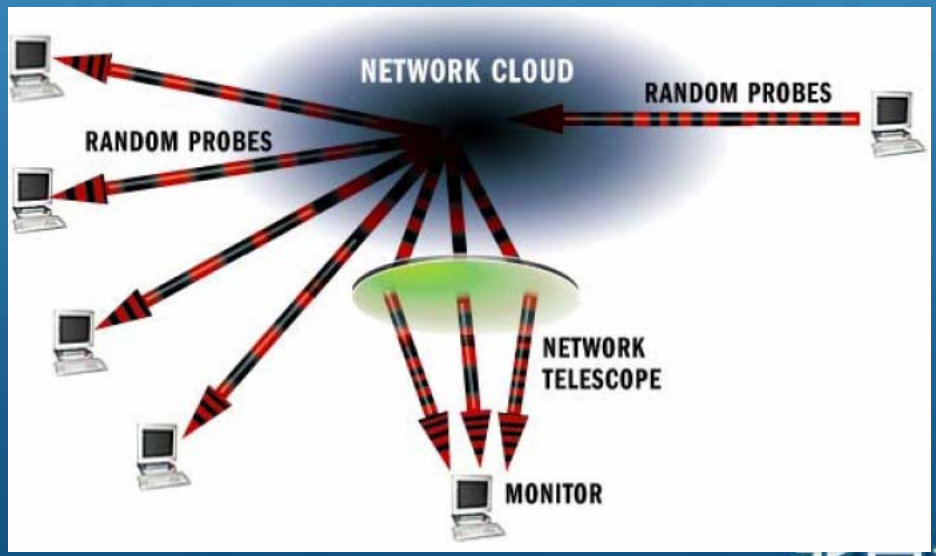
# 零交互式蜜罐技术

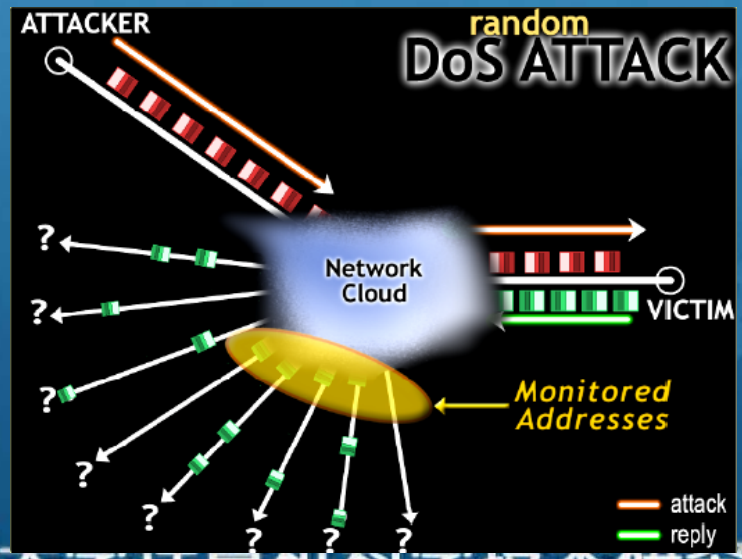
- 零交互式蜜罐技术
  - 利用未分配/未使用的IP地址资源（“黑洞”）
  - 前提假设：任何流向未使用IP地址的数据包都预示着攻击或异常
  - 零交互式：与数据包源没有任何交互，只是被动监听
- 零交互式蜜罐工具
  - Network Telescope - CAIDA
  - Darknet Project - Team Cymru



# Network Telescope

- 随机扫描攻击（恶意代码扫描传播）
  - Code Red, Slammer, Witty, …
- 伪造源地址DDoS攻击—Backscatter Analysis （未请求反向包分析）
- 统计分析结论
  - 使用较大的IP地址块（/16-/8）对全局安全实时监测非常重要
  - 使用分布的多地址块进行监测能够大幅提高监测效果



# 低交互式蜜罐技术

- 低交互式蜜罐技术
  - 具有与攻击源主动交互的能力
  - 模拟网络服务响应，模拟漏洞
  - 容易部署，容易控制攻击
  - 低交互式—交互级别由于模拟能力而受限，数据获取能力和伪装性较弱，一般仅能捕获已知攻击
- 低交互式蜜罐工具
  - iSink – 威斯康星州立大学
  - Internet Motion Sensor – 密歇根大学，Arbor Networks
  - Honeyd — Google公司软件工程师Niels Provos
  - Nepenthes – 德国蜜网项目组Nepenthes团队
  - 商业产品: KFSensor, Specter, HoneyPoint...

# 高交互式蜜罐技术

- 高交互式蜜罐技术
  - 使用真实的操作系统、网络服务与攻击源进行交互
  - 高度的交互等级—对未知漏洞、安全威胁具有天然的可适性，数据获取能力、伪装性均较强
  - 弱势—资源需求较大，可扩展性较弱，部署安全风险较高
- 虚拟机蜜罐 VS. 物理蜜罐
  - 虚拟机(Virtual Machine)/仿真器(Emulator)技术
  - 节省硬件资源、容易部署和控制、容易恢复、安全风险降低
- 高交互式蜜罐工具
  - Honeynet – 蜜网项目组 (The Honeynet Project)
  - HoneyBow (基于高交互式蜜罐的恶意代码捕获器) – 北京大学狩猎女神项目组
  - Argos – 荷兰阿姆斯特丹大学 (Vrije Universiteit Amsterdam) 欧盟分布式蜜罐项目(NoAH)参与方

# 蜜罐技术的另一分类维度

- 按蜜罐的应用目标分类
- 产品型蜜罐
  - 目标是有效防护业务网络
  - 间接性防护—通过诱骗增大攻击者代价，混淆关键业务资源，了解并规避安全威胁
  - 直接性防护—蜜场技术
  - KFSensor, Specter, Symantec Decoy Server, ...
- 研究型蜜罐
  - 目标是研究对手，了解自身面临的安全威胁
  - 知己知彼、百战不殆
  - 蜜网技术(Know Your Enemy)—目前更多意义上属于研究型蜜罐技术
  - Honeyd, Honeynet, ...

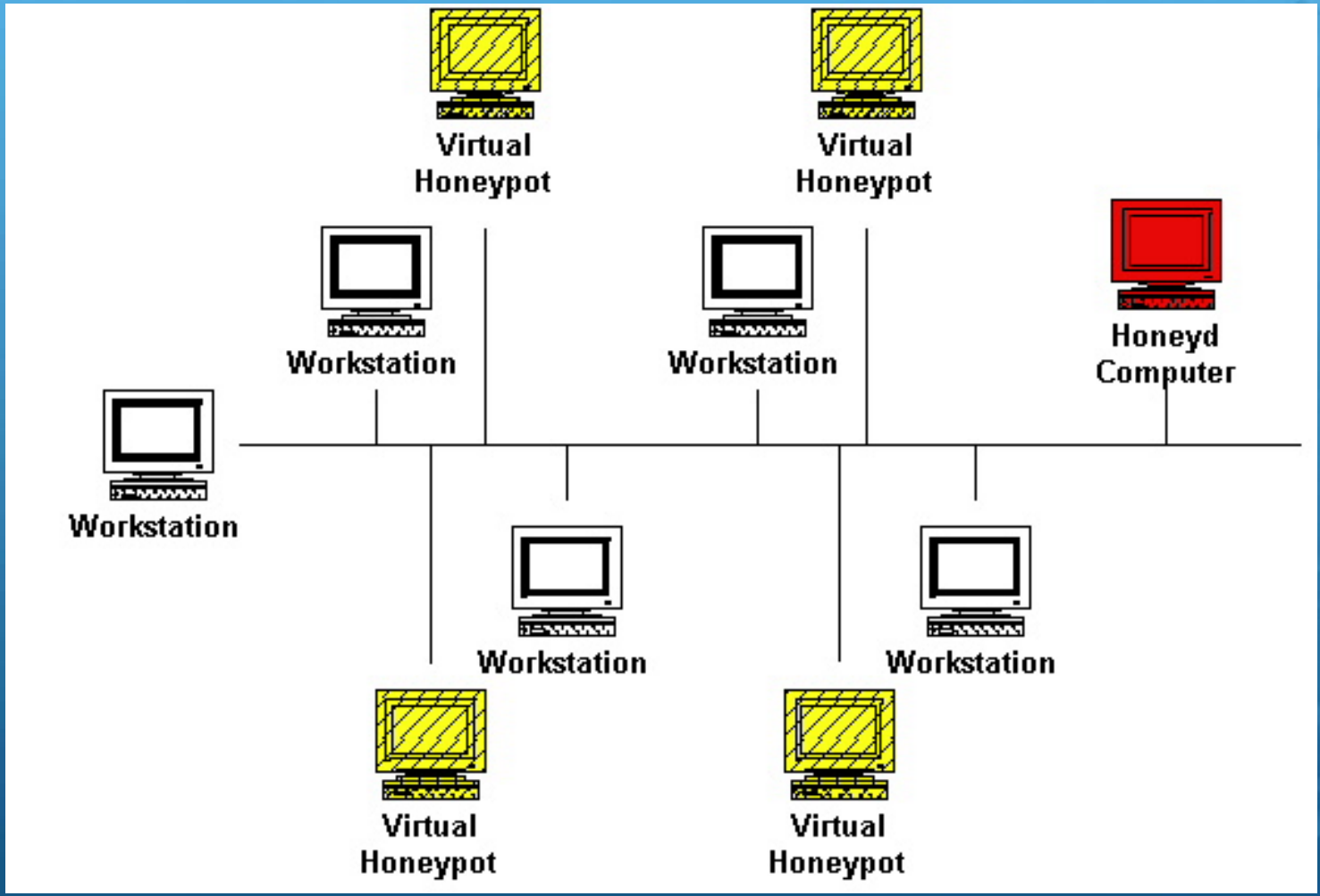
# 蜜罐技术

- 蜜罐技术的提出与发展历程
- 蜜罐技术的概念和原理
- 蜜罐技术的分类
- 典型蜜罐工具
  - Honeyd (*nix平台)
  - KFSensor (Win32平台)

# Honeyd

- A virtual honeypot framework
  - by Niels Provos, Google Inc.
  - 最新版本：1.5b
- 支持同时模拟多个IP地址主机
  - 经过测试，最多同时支持65535个IP地址
  - 支持模拟任意的网络拓扑结构
- 通过服务模拟脚本可以模拟任意TCP/UDP网络服务
  - IIS, Telnet, pop3...
- 支持ICMP
  - 对ping和traceroutes做出响应
- 通过代理机制支持对真实主机、网络服务的整合
  - add windows tcp port 23 proxy “162.105.204.159 23”

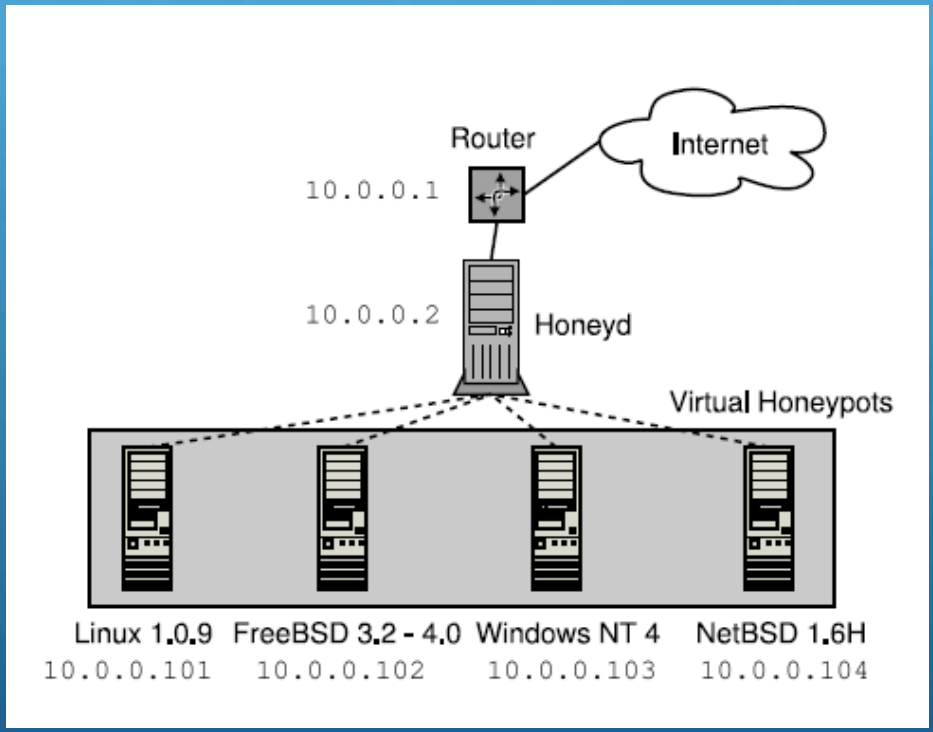
# Honeyd监控未使用IP地址



# Honeyd设计上的考虑

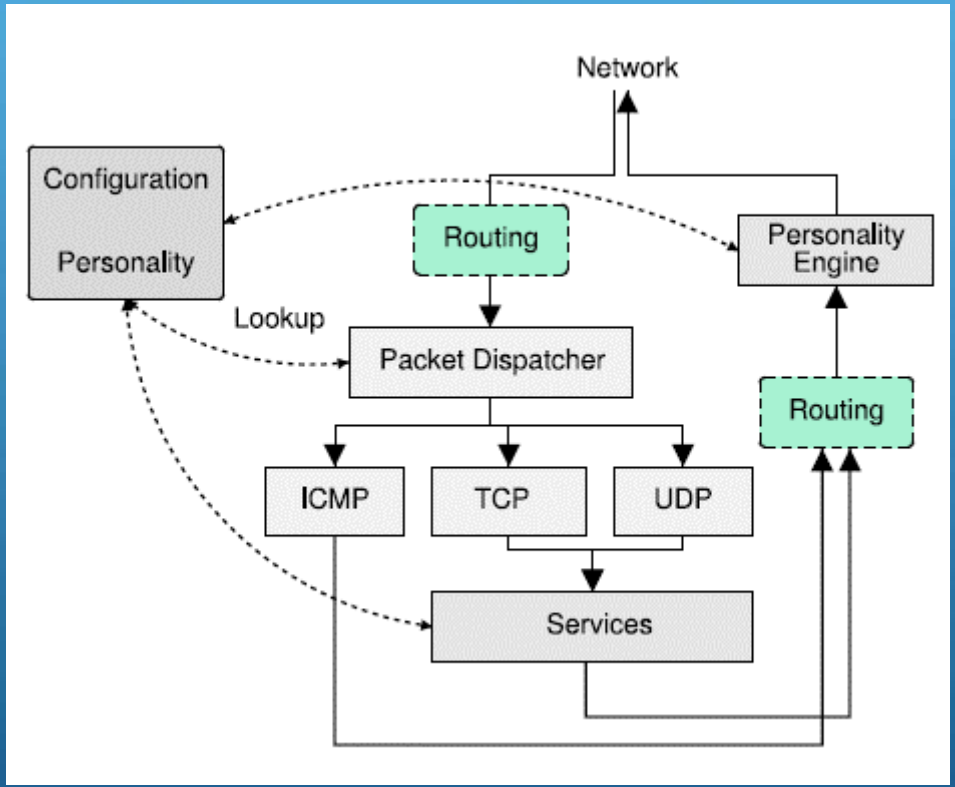
- 接收网络流量
- 模拟蜜罐系统
  - 仅模拟网络协议栈层次，而不涉及操作系统各个层面
  - 可以模拟任意的网络拓扑
- Honeyd宿主主机的安全性
  - 限制只能在网络层面与蜜罐进行交互
- 捕获网络连接和攻击企图
  - 日志功能

# 接收网络流量




- Honeyd模拟的蜜罐系统接收相应网络流量三种方式
  - 为Honeyd模拟的虚拟主机建立路由
  - ARP代理
  - 支持网络隧道模式 (GRE)

# Honeyd体系框架




- 路由模块
- 中央数据包分发器
  - 将输入的数据包分发到相应的协议处理器
- 协议处理器
  - Service模拟脚本
- 个性化引擎
- 配置数据库
  - 存储网络协议栈的个性化特征

# 路由模块

- Honeyd支持创建任意的网络拓扑结构
  - 对路由树的模拟
    - 配置一个路由进入点
    - 可配置链路时延和丢包率
    - 模拟任意的路由路径
  - 扩展
    - 将物理主机融合入模拟的网络拓扑
    - 通过GRE隧道模式支持分布式部署

# FTP服务模拟脚本

```
case $incmd_nocase in

  QUIT* )
        echo -e "221 Goodbye.\r"
        exit 0;;
  SYST* )
        echo -e "215 UNIX Type: L8\r"
        ;;
  HELP* )
        echo -e "214-The following commands are recognized (*=>'s unimplemented).\r"
        echo -e "   USER    PORT    STOR    MSAM*   RNTO    NLST    MKD     CDUP\r"
        echo -e "   PASS    PASV    APPE    MRSQ*   ABOR    SITE    XMKD    XCUP\r"
        echo -e "   ACCT*   TYPE    MLFL*   MRCP*   DELE    SYST    RMD     STOU\r"
        echo -e "   SMNT*   STRU    MAIL*   ALLO    CWD     STAT    XRMD    SIZE\r"
        echo -e "   REIN*   MODE    MSND*   REST    XCWD    HELP    PWD     MDTM\r"
        echo -e "   QUIT    RETR    MSOM*   RNFR    LIST    NOOP    XPWD\r"
        echo -e "214 Direct comments to ftp@$domain.\r"
        ;;
  USER* )
```

# 个性化引擎

- 为什么需要个性化引擎?
  - 不同的操作系统有不同的网络协议栈行为
  - 攻击者通常会运行指纹识别工具，如Xprobe和Nmap获得目标系统的进一步信息
  - 个性化引擎使得虚拟蜜罐看起来像真实的目标
- 每个由Honeyd产生的包都通过个性化引擎
  - 引入操作系统特定的指纹，让Nmap/Xprobe进行识别
  - 使用Nmap指纹库作为TCP/UDP连接的参考
  - 使用Xprobe指纹库作为ICMP包的参考

# 日志功能

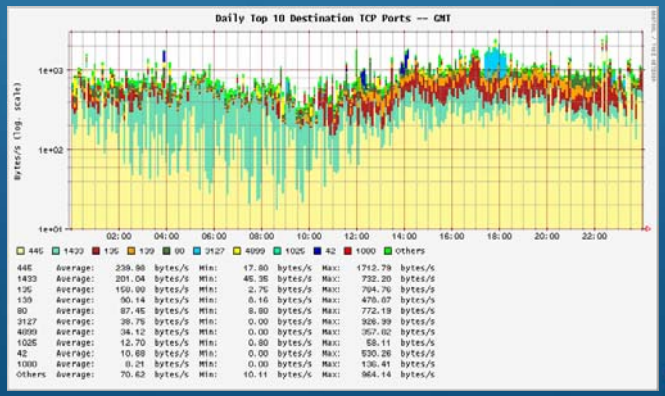
- Honeyd的日志功能
  - Honeyd对任何协议创建网络连接日志，报告试图发起的、或完整的网络连接
  - 在网络协议模拟实现中可以进行相关信息收集

```
Feb 12 23:06:33 Connection to closed port: udp (210.35.128.1:1978 -
172.16.85.101:1978)
Feb 12 23:23:40 Connection request: tcp (66.136.92.78:3269 - 172.16.85.102:25)
Feb 12 23:23:40 Connection established: tcp (66.136.92.78:3269 - 172.16.85.102:25)
<-> sh scripts/smtp.sh
Feb 12 23:24:14 Connection dropped with reset: tcp (66.136.92.78:3269 -
172.16.85.102:25)
Feb 12 23:34:53 Killing attempted connection: tcp (216.237.78.227:3297 -
172.16.85.102:80)

Wed Feb 12 23:23:40 UTC 2003: SMTP started from  Port
EHLO relay.verizon.net
```

# Honeyd的应用

- 反蠕虫
  - Snipe Blaster: add default tcp port 4444 "/bin/sh scripts/strikeback.sh $ipsrc"
- Honeycomb – SIGCOMM paper
  - 自动生成NIDS检测特征
- 巴西蜜罐联盟－安全监测



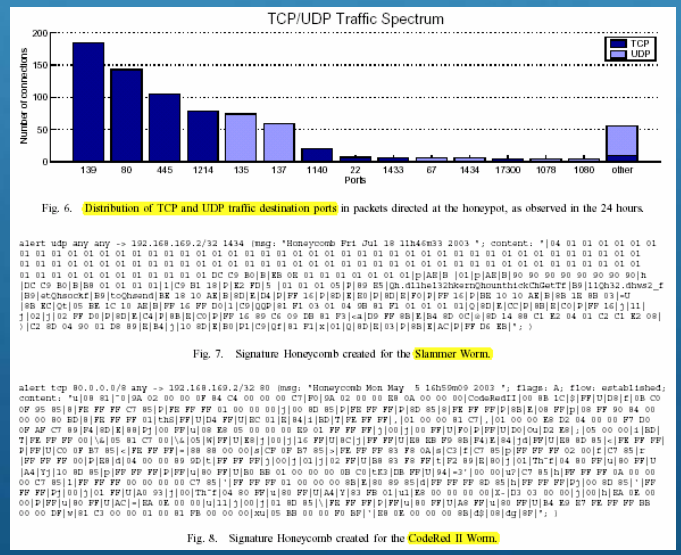




Fig. 6. Distribution of TCP and UDP traffic destination ports in packets directed at the honeypot, as observed in the 24 hours.

```
alert udp any any -> 192.168.169.2/32 1434 (msg: "Honeycomb Fri Jul 18 11h46m33 2003 "; content: "|04 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 DC C9 B0|B|EB 0E 01 01 01 01 01 01 01|p|AE|B |01|p|AE|B |90 90 90 90 90 90 90 90|h
|DC C9 B0|B|B8 01 01 01 01|1|C9 B1 18|P|E2 FD|5 |01 01 01 05|P|89 E5|h.dllhel32hkernQhounthickChGetTf|B9|11Qh32.dhws2_f
|B9|atQhsockf|B9|toQhsend|BE 18 10 AE|B|8D|E|D4|P|FF 16|P|8D|E|B0|P|8D|E|F0|P|FF 16|P|BE 10 10 AE|B|8B 1E 8B 03|=U
|8B EC|Qt|05 BE 1C 10 AE|B|FF 16 FF D0|1|C9|QQP|81 F1 03 01 04 5B 81 F1 01 01 01|Q|8D|E|2C|P|B8|E|C0|P|FF 16|j|1|
j|02|j|02 FF D0|P|8D|E|C4|P|8B|E|C0|P|FF 16 89 C6 09 DB 81 F3|<a|D9 FF 8B|E|B4 8D 0C|e|8D 14 8B C1 E2 04 01 C2 C1 E2 08|
 |C2 8D 04 90 01 D8 89|E|B4|j|10 8D|E|B0|P1|C9|Qf|81 F1|x|01|Q|8D|E|03|P|8B|E|AC|P|FF D6 EB|*; )
```

Fig. 7. Signature Honeycomb created for the Slammer Worm.

```
alert tcp 80.0.0.0/8 any -> 192.168.169.2/32 80 (msg: "Honeycomb Mon May  5 16h59m09 2003 "; flags: A; flow: established;
content: "u|08 81|^0|0A 02 00 00 0F 84 C4 00 00 00 C7|P0|0A 02 08 00 E8 0A 00 00 00|CodeRedII|00 8B 1C|$|FF|U|D8|f|8B C0
0F 95 85|8|FE FF FF C7 85|P|FE FF FF 01 00 00 00|j|00 8D 85|P|FE FF FF|P|8D 85|8|FE FF FF|P|8B|E|08 FF|p|08 FF 90 84 00
00 00 80 8D|8|FE FF FF 01|th8|FF|U|D4 FF|U|EC 01|E|84|1|8D|7|FE FF FF|.|01 00 00 01 C7|.|01 00 00 E8 E2 04 00 00 F7 D0
0F AF C7 89|F4|8D|E|88|P|00 FF|u|08 E8 05 00 00 00 E9 01 FF FF FF|j|00|j|00 FF|U|F0|P|FF|U|D0|Ou|D2 E8|.|05 00 00|i|8D|
7|FE FF FF 00|\&|05 81 C7 00|\&|05|W|FF|U|E8|j|00|j|16 FF|U|8C|j|FF FF|U|E8 EB F9 8B|F4|E|84|jd|FF|U|E8 8D 85|<|FE FF FF|
P|FF|U|C0 0F B7 8S|<|FE FF FF|=|88 88 00 00|s|CF 0F B7 8S|>|FE FF FF 83 F8 0A|s|C3|f|C7 85|p|FF FF FF 02 00|f|C7 85|r
|FF FF FF 00|P|E8|d|04 00 00 89 9D|t|FF FF FF|j|00|j|01|j|02 FF|U|B8 83 F8 FF|t|F2 89|E|80|j|01|Th*f|04 80 FF|u|80 FF|U
|A4|Y|10 8D 85|p|FF FF FF|P|FF|u|80 FF|U|B0 BB 01 00 00 00 0B C9|tE|EB FF|U|84|=2'|00 00|u7|C7 85|h|FF FF FF 0A 00 00
00 C7 85|1|FF FF FF 00 00 00 00 C7 85|*|FF FF FF 01 00 00 00 8B|E|80 89 85|d|FF FF FF 8D 85|h|FF FF FF|P|00 8D 85|*|FF
FF FF|P|00|j|01 FF|U|A0 93|j|00|Th*f|04 80 FF|u|80 FF|U|A4|Y|83 F8 01|u1|E8 00 00 00 00|X-|D3 03 00 00|j|00|h|EA 0E 00
00|P|FF|u|80 FF|U|AC|=|EA 0E 00 00|u|11|j|00|j|01 8D 85|\|FE FF FF|P|FF|u|80 FF|U|A8 FF|u|80 FF|U|B4 E0 E7 FE FF FF B8
00 00 DF|w|81 C3 00 00 01 00 81 FB 00 00 00|xu|05 B8 00 00 FO BF|'|E8 0E 00 00 00 8B|dj|08|dg|8F|*; )
```

Fig. 8. Signature Honeycomb created for the CodeRed II Worm.

# 蜜罐技术

- 蜜罐技术的提出与发展历程
- 蜜罐技术的概念和原理
- 蜜罐技术的分类
- 典型蜜罐工具
  - Honeyd (*nix平台)
  - KFSensor (Win32平台)

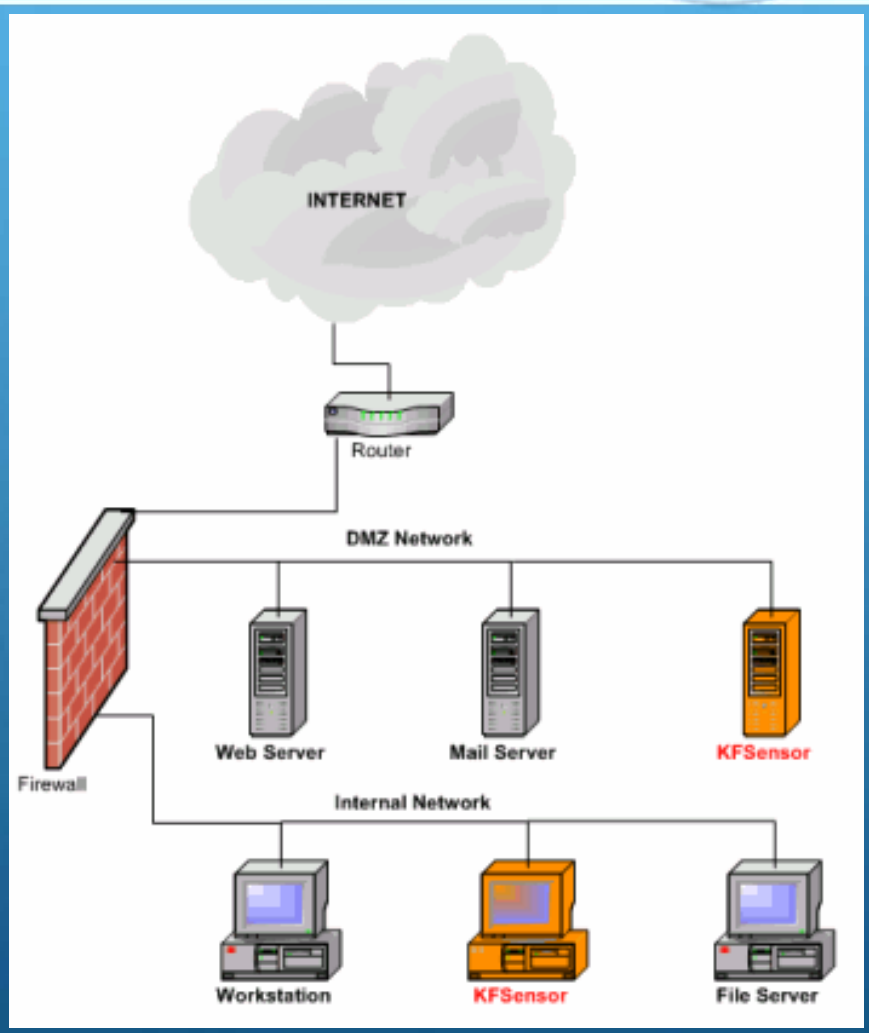
# KFSensor介绍

- KeyFocus公司推出的商业低交互式蜜罐产品
  - 2002年推出，目前最新版本为4.2.0
  - 提供两周试用版
  - 依赖于winPcap库
  - 支持模拟的服务
    - TCP, UDP, ICMP
    - Sim Banner：记录网络数据，返回Banner标识
    - Sim Stand Server：模拟真实服务
  - 模拟服务的所属类
    - Windows工作站、服务器、互联网服务、Linux、木马和蠕虫后门服务

| 服务 | 端口 |
|---|---|
| CMD Shell | 4444 |
| DHCP | 67 |
| FTP | 21 |
| HTTP | 80 |
| NBT Name | 137 |
| NBT Data | 138 |
| NBT Session | 139 |
| NBT SMB | 445 |
| POP3 | 110 |
| Relay | |
| SMTP | 25 |
| SOCKS | 1080 |
| SQL | 1433 |
| SQL UDP | 1434 |
| Telnet | 23 |
| Terminal | 3389 |
| VNC | 5900 |

# KFSensor的部署

- 部署目标
  - 检测攻击
  - 增加攻击代价
  - 为应急响应提供丰富信息
  - 研究
- 部署位置
  - 直接接入互联网
  - DMZ非军事区
  - 内部网络
  - 攻击检测、诱骗网络



# KFSensor关键概念

- 访问者(Visitor)
  - 黑客、恶意代码、偶然闯入的合法用户
- 模拟服务(Sim Server)
  - 包括Sim Banner和Sim Stand Server两类
- 事件(Event)
  - KFSensor探测到的访问者对模拟服务的一次访问事件
- 监听动作(Listen)
  - 直接关闭、读取访问者的请求并关闭、由模拟服务进行响应
- 场景(Scenario)
  - 定义了KFSensor一组开放模拟服务，以及对事件的处理动作
- 报警(Alert)
  - 系统托盘，声音，电子邮件，Syslog，事件日志，外部报警
- 特征码(Signature)—标识已知攻击
- 防止DoS攻击—能够检测并应对DoS攻击

# 休息、提问时间

10分钟

# 演示：典型蜜罐工具

*nix平台－Honeyd

Win32平台－KFSensor