Dionaea 低交互式蜜罐介绍

代恒，诸葛建伟
(清华大学网络中心，CCERT 应急响应组)

1. Dionaea(捕蝇草)低交互式蜜罐简介

Dionaea 低交互式蜜罐(http://dionaea.carnivore.it/）是 Honeynet Project 的开源项目，起始于 Google Summer of Code 2009，是 Nepenthes(猪笼草)项目的后继。Honeynet Project 是成立于 1999 年的国际性非盈利研究组织，致力于提高因特网的安全性，在蜜罐技术与互联网安全威胁研究领域具有较大的影响力。

Dionaea 蜜罐的设计目的是诱捕恶意攻击，获取恶意攻击会话与恶意代码程序样本。它通过模拟各种常见服务，捕获对服务的攻击数据，记录攻击源和目标 IP、端口、协议类型等信息，以及完整的网络会话过程，自动分析其中可能包含的 shellcode 及其中的函数调用和下载文件，并获取恶意程序。

有别于高交互式蜜罐采用真实系统与服务诱捕恶意攻击，Dionaea 被设计成低交互式蜜罐，它为攻击者展示的所有攻击弱点和攻击对象都不是真正的产品系统，而是对各种系统及其提供的服务的模拟。这样设计的好处是安装和配置十分简单，蜜罐系统几乎没有安全风险，不足之处是不完善的模拟会降低数据捕获的能力，并容易被攻击者识别。

2. Dionaea 的工作机制

2.1. 技术原理和整体结构

Dionaea 是运行于 Linux 上的一个应用程序，将程序运行于网络环境下，它开放 Internet 上常见服务的默认端口，当有外来连接时，模拟正常服务给予反馈，同时记录下出入网络数据流。网络数据流经由检测模块检测后按类别进行处理，如果有 shellcode 则进行仿真执行；程序会自动下载 shellcode 中指定下载或后续攻击命令指定下载的恶意文件。从捕获数据到下载恶意文件，整个流程的信息都被保存到数据库中，留待分析或提交到第三方分析机构。

Dionaea 整体结构和工作机制如图：

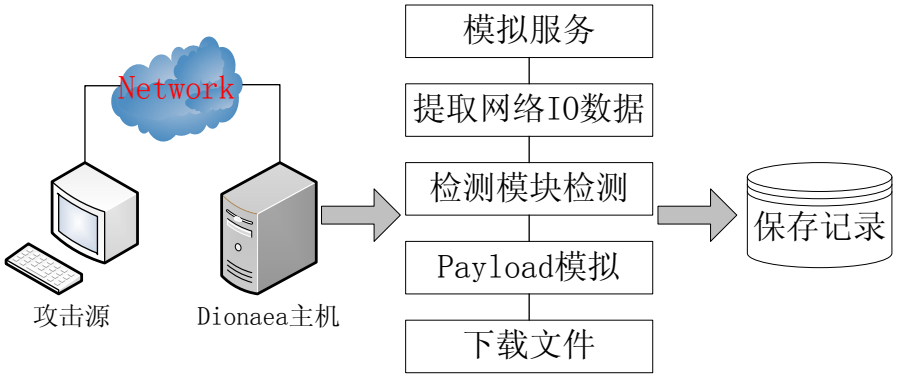

图 1 Dionaea 整体结构和工作机制

2.2. 模拟服务

Dionaea 通过模拟常见的 Internet 服务引诱网络上以这些服务为目标的攻击，因为蜜罐系统其实并不提供任何服务，可以假定所有服务请求都是恶意的。目前支持的协议类型和服务有 SMB、http、ftp、tftp、MSSQL、MySQL、SIP(VoIP)，同时还支持 IPv6 和 TLS（Transport Layer Security，传输层安全）。

按默认配置启动蜜罐，程序自动获取网络接口 IP 地址，在 IPv4 和 IPv6 同时开启监听服务。服务开放的端口，tcp 端口对应有 web 服务 80、443 端口，ftp 服务 21 端口，MSSQL 的 1433 端口，MySQL 的 3306 端口，支持 SMB 的 445 端口，RPC（Remote Procedure Call，远程过程调用）和 DCOM（分布式组件对象模型）服务使用的 135 端口，wins 服务 42 端口；udp 端口有 VoIP 使用的 SIP（Session Initiation Protocol，会话发起协议）对应的 5060 端口，tftp 服务 69 端口。新的服务可以通过编写 python 脚本的方式添加到蜜罐中，具有很强的扩展性。MySQL 服务即为 2011 年 5 月发布的新版本中最新添加的模拟服务。

2.3. 捕获攻击数据

对于模拟服务的监听连接，Dionaea 直接获取外来连接数据，然后根据模拟服务的实现向外返回数据；从没有模拟服务支持的端口收到的连接包则会先被记录，然后丢弃。获取的外来数据可以提交给检测模块进行检测，同时有关此连接的信息，如源 IP、目的 IP、源端口、目的端口、协议类型等会被记录到数据库中，方便进行分析和统计。

2.4. Shellcode 检测和仿真引擎

外来数据提交到检测模块后，若检测到 shellcode，就被放到程序自带的虚拟机中进行仿真执行。检测和仿真引擎采用 libemu，同样是由 Honeynet Project 开发的 x86 下 shellcode 检测和仿真程序库。它采用 GetPC 启发式模式来检测数据流中是否有 shellcode，发现后在虚拟机中运行代码，并记录 API 调用和参数，对于多级 shellcode 同样可以仿真。

Libemu 按字节检查输入数据，发现连续的机器指令则提取出来，交给程序的虚拟机进行仿真。以 metasploit 中的 payload “download_exec”为例，Dionaea 收到溢出数据流并处理后，在下载参数中发现 url 字符串，则利用 curl 程序到此 url 下载文件并保存。以 metasploit 中的 payload “shell_bind_tcp”为例，设置参数为开放 4444 端口后门，Dionaea 会开放 4444 端口并等待进一步的连接数据，如黑客连接后门的操作，程序会对命令做相应反馈以吸引更多输入数据，当然 Dionaea 对后门的模拟与高交互式蜜罐相比差距很大。

2.5. 恶意文件下载

Dionaea 蜜罐会自动检测 shellcode 中的文件下载地址或黑客指定的下载地址，并尝试下载文件。http 下载利用 curl 模块，而 ftp 下载和 tftp 下载利用的是程序自带的 ftp.py 和 tftp.py 这两个 python 脚本。

2.6. 事件记录

默认安装目录为/opt/dionaea/，当有针对模拟服务的恶意攻击发生时，一系列出入数据会被记录下来，保存到/opt/dionaea/var/dionaea/bistreams 目录下按时间命名的目录中。文件下载成功后，会被保存到/opt/dionaea/var/dionaea/binaries 目录下。而用户向模拟服务上传的文件将被保存在/opt/dionaea/var/dionaea/wwwroot 目录下。

检测到的事件信息将被保存到事件数据库中。Dionaea 使用的是 SQLite 数据库，数据库文件 logsql.sqlite 默认位于/opt/dionaea/var/dionaea/目录下。

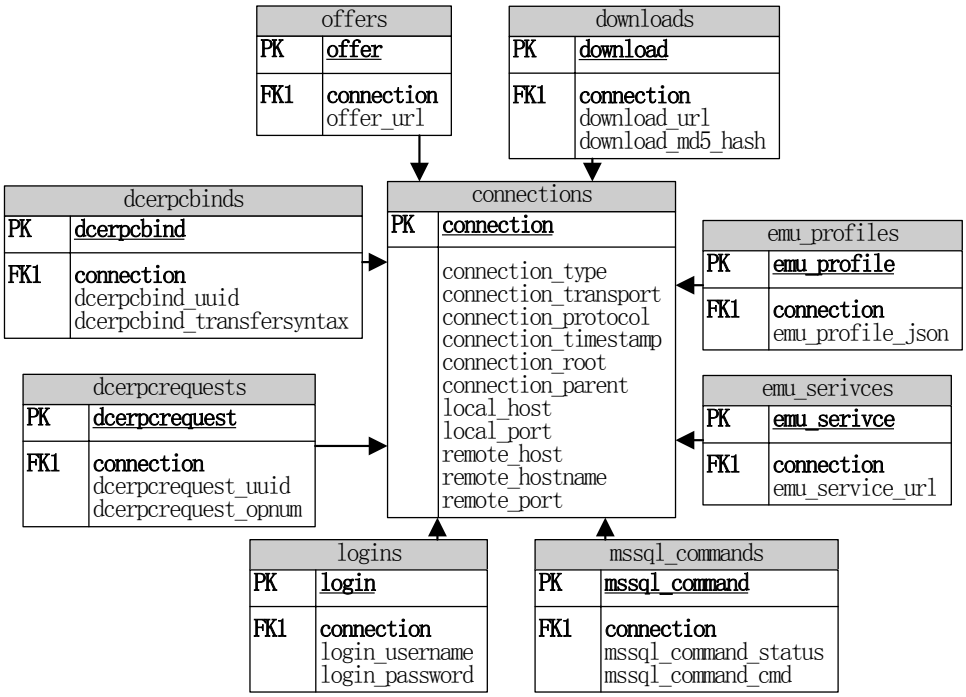

图2 事件数据库主要结构

图 2 展示了 Dionaea 事件数据库主要的几个表和关系结构。数据库以 connections 表为核心，记录了外部攻击和蜜罐的每一次连接行为，包括 IP、端口、协议类型和时间戳等信息。其中 connection_root 键记录引起本次连接的最初一个连接编号，connection_parent 键记录引起本次连接的上一个连接编号，通过这两个键可以将一次连续的攻击提取出来。如连接 Link1 溢出攻击的 shellcode 控制主机连接到攻击控制端，新建连接 Link2，则 Link1 是 Link2 的“root”和“parent”。

以 connections 表为轴，其他表记录详细攻击行为信息。dcerpcbinds 表和 dcerpcrequests 表当连接使用 SMB 协议时会产生记录，其中 UUID 标识调用的分布式远程过程，两表分别记录调用所采用的编码规则和请求操作号(Operation number)。emu_profiles 表和 emu_serivces 表记录 libemu 程序对数据流中 shellcode 检测得到的结果，分别记录检测到的 shellcode 函数调用、参数和判断出的 shellcode 开启的后门。downloads 表记录了下载的恶意文件的信息，包括下载地址和对文件的 md5 值。offers 表记录 shellcode 中提供的文件下载 url。logins 表和 mssql_commands 表记录对 MSSQL 服务的攻击行为，分别记录尝试登陆的用户名、密码和对数据库提交的命令以及命令完成情况。

3. Dionaea 蜜罐安装过程

Dionaea 目前版本是 0.1.0，采用源码安装。软件运行依赖于以下库：libev，libglib，libssl，liblcfg，libemu，python，sqlite，readline，cython，lxml，libudns，libcurl，libpcap。安装过程详见 http://dionaea.carnivore.it/#compiling，需要注意的是安装 Python-3.2 时注意按说明修改 setup.py 以确保 zlib 库能正确安装。

安装时要注意依赖库成功安装，否则 Dionaea 可能不能正常工作。网上也有提供利用脚本整合安装的方法，具体可参考以下网址：

http://carnivore.it/2010/05/18/debianization

http://aur.archlinux.org/packages.php?ID=36944

http://maimed.org/~pyllyukko/stuff.shtml

4. Dionaea 使用方法

Dionaea 根据命令参数运行，可选择不同的运行环境、任务和筛选事件记录内容。配置文件则具体规定蜜罐运行后开启的模块，记录文件的保存位置和扩展功能的参数等信息。默认配置下 Dionaea 自动选择一个网络接口进行监听。

4.1. Dionaea 的命令格式

Dionaea 具体的命令格式如下：

dionaea [**-c**, --config=FILE] [**-D**, --daemonize] [**-g**, --group=GROUP]
[**-G**, --garbage=[collect|debug]] [**-h**, --help] [**-H**, --large-help]
[**-l**, --log-levels=WHAT] [**-L**, --log-domains=WHAT] [**-u**, --user=USER]
[**-p**, --pid-file=FILE] [**-r**, --chroot=DIR] [**-V**, --version] [**-w**, --workingdir=DIR]

选项的意义分别是：

-c：指定运行程序所使用的配置文件，默认下配置文件是/opt/dionaea/etc/dionaea.conf。

-D：后台运行。

-g：指定启动后切换到某个用户组，默认下保持当前组。

-G：收集垃圾数据，用于调试内存泄露。不能用于 valgrind 软件。

-h：帮助信息。

-H：帮助信息，包括默认值信息。

-l：选择事件记录级别，可以选择 all, debug, info, message, warning, critical, error 这些值，多选使用“,”做分隔，排除使用“-”。

-L：选择域，支持通配符“*”和“?”，多选使用“,”，排除使用“-”。

-u：指定启动后切换到某个用户，默认下保持当前用户。

-p：记录 pid 到指定文件。

-r：指定启动后切换根目录到指定目录，默认下不切换。

-V：显示版本信息。

-w：设定进程工作目录，默认下为/opt/dionaea。

4.2. Dionaea 配置文件

Dionaea 默认下配置文件是/opt/dionaea/etc/dionaea.conf。配置文件内容分为 logging，processors，downloads，bistreams，submit，listen，modules 七个部分。

日志 logging 部分配置日志的存放位置、事件记录级别和所在域，包括普通日志和错误日志，默认下位于/opt/dionaea/var/log 目录下，分别记录所有事件和警告、错误事件。

处理器 processors 部分配置 libemu 和用于导出数据流的模块 streamdumper。libemu 部分可增减允许的协议，配置 shellcode 检测时支持的最大流大小、跟踪步数限制和并发执行数等性能参数。streamdumper 部分配置导出数据流时允许和拒绝的协议，数据流保存的位置。

文件下载 downloads 和数据流 bistreams 部分分别配置恶意文件下载和数据流保存的位置。

提交 submit 部分设置自动通过 http 提交恶意文件到特定地址，具体配置信息依赖于服务器的设定。

监听设置 listen 部分配置 Dionaea 进行监听的网络接口 IP，默认下自动获取。

工作模块 modules 部分配置各种模块的工作参数。部分必须模块，如 curl、libemu、pcap、模拟的服务 services 等信息，建议保存默认配置。其他可供扩展的模块将在第五部分 Dionaea 进阶使用高级功能进行介绍。

4.3. Dionaea 使用实例

Dionaea 安装完成后可按需求进行一定配置，但通常使用默认配置也可实现决大多数功能。下面对 Dionaea 默认配置下的使用和工作情况作简单介绍。

以下 Dionaea 0.1.0 安装于 Ubuntu 10.10，测试机为 BT5（back track 5）。启动 Dionaea：dionaea -l all,-debug -L '*' 。记录除 debug 之外所有事件，工作于所有可见域。

4.3.1. 模拟的服务

```
[26072011 14:01:03] connection connection.c:3847: connection 0x8b0c6f0 none/tcp
type: none->accept
[26072011 14:01:03] connection connection.c:3880: connection 0x8b0c6f0 accept/tc
p/none [192.168.255.132:80->192.168.255.1:4274] state: none->established
[26072011 14:01:03] logsql dionaea/logsql.py:547: accepted connection from 192.1
68.255.1:4274 to 192.168.255.132:80 (id=16)
[26072011 14:01:13] connection connection.c:3880: connection 0x8b0c6f0 accept/tc
p/established [192.168.255.132:80->192.168.255.1:4274] state: established->close
[26072011 14:01:13] logsql dionaea/logsql.py:612: attackid 16 is done
[26072011 14:01:32] logsql dionaea/logsql.py:560: reject connection from 192.168
.255.1:4277 to 192.168.255.132:81 (id=17)
[26072011 14:01:32] logsql dionaea/logsql.py:612: attackid 17 is done
```

图3 nc 连接80和81端口 Dionaea 蜜罐输出

图3显示使用 nc（netcat）连接 Dionaea 开放端口（80）和未开放端口（81）时蜜罐的处理情况。连接端口80（蓝色框内）时 Dionaea 接受连接，根据应用服务模拟模块中的实现提供低交互式的模拟交互环境，并将记录网络会话过程；而连接81（红色框内）端口时连接被 Dionaea 拒绝，仅通过 pcap 模块记录连接尝试。

4.3.2. 漏洞扫描

使用漏洞扫描软件扫描 Dionaea 蜜罐，可以看到软件识别出多个安全漏洞，虽然这些漏洞都不实际存在。使用 Nessus 扫描时，共扫描出 5 个高危漏洞、13 个中等威胁漏洞和 28 个低威胁漏洞，而用 OpenVAS 进行扫描，选用模式“Full and fast”，扫描出 1 个高危漏洞、1 个中等威胁漏洞和 17 个低威胁漏洞。

从扫描结果可以看出，Dionaea 虽然可以捕获针对漏洞的攻击数据，但在模拟漏洞吸引攻击方面还做的不够，更多的漏洞模拟理论上可以捕获更多的攻击数据。当然，这是低交互式蜜罐普遍存在的问题，需待以后进一步研究。

4.3.3. 事件记录的读取

Dionaea 提供两个 python 脚本用于事件数据库的简单读取查看，脚本在安装源码保存位置的 modules/pythom/目录下。

脚本 readlogsqltree.py 可以读取数据库文件 logsql.sqlite 中保存的攻击连接信息，包括网络连接信息、时间、针对的漏洞、shellcode 调用的 API 和参数、下载文件的 url，若配置过 virustotal 提交分析，还可以显示提交分析的结果，如下命令格式输出24小时内的事件记录：

./readlogsqltree.py -t $(date '+%s')-24*3600 /opt/dionaea/var/dionaea/logsql.sqlite

脚本 gnuplotsql.py 统计事件数据库中记录的连接数，然后输出统计图形。可以按日期统计不同协议、不同连接类型的连接数。使用前需安装 gnuplot 程序：

apt-get install gnuplot

下面的命令执行后，将生成所有连接和 smbd 、epmapper、mssqld、httpd、ftpd 五种协议的连接统计图，同时输出按年和月统计的图形：

./gnuplotsql.py -d /opt/dionaea/var/dionaea/logsql.sqlite -p smbd -p epmapper -p mssqld -p httpd -p ftpd

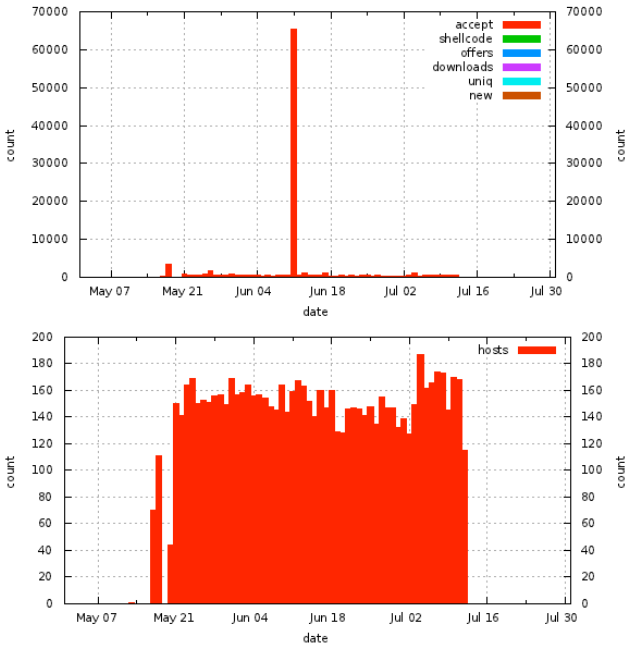

图4 校外服务器数据库连接统计图示例

还可以直接使用能读取 sqlite 格式数据库文件的软件查看事件记录。如 windows 下的软件 SQLiteSpy。

5. Dionaea 进阶使用高级功能

除默认功能外，Dionaea 还提供一些非常实用的扩展功能，或者增强信息获取能力，或者帮助对恶意攻击和恶意文件进行分析。

5.1. 配置 p0f 程序提供远程主机 OS 信息

程序 p0f 用于通过网络连接被动识别远程操作系统指纹。在配置文件中工作模块 modules 部分的 ihandlers 结构中开启 p0f，可以获取连接的远程操作系统信息。
使用此功能需要预先安装 p0f：apt-get install p0f。使用时，先以命令“sudo p0f -i any -u root -Q /tmp/p0f.sock -q -l”启动 p0f，之后启动 Dionaea 蜜罐。当有连接进入时就能获取 p0f 判断的操作系统信息，此信息同时保存到数据库中。

5.2. 多网络接口支持

Dionaea 可安装于多网卡主机，同时监听多个网络接口。默认情况下 Dionaea 自动获取监听 IP，存在多 IP 时，需要更改配置文件 listen 部分为手动填写 IP，同时添加所需多个 IP。举例如下：

```
listen =
{
mode = “manual”#手动输入监听 IP
addrs = { eth0 = ["192.168.0.1", "192.168.0.2", "192.168.0.3", "192.168.0.4"] }
}
```

5.3. Dionaea 的样本扫描和分析

Dionaea 支持以 http/POST 形式将下载的恶意文件自动提交到第三方分析机构，如 VirusTotal，CWSandbox，Norman Sandbox，也可提交到自己配置的服务器。

VirusTotal 提供对提交的文件进行多引擎扫描服务。Dionaea 对 VirusTotal 做了支持，设置提交恶意文件到 VirusTotal，首先要到网站 **http://www.virustotal.com** 注册，获取网站提供的 apikey，再把 apikey 写到配置文件中工作模块 modules 部分的 virustotal 结构中，同样到 ihandlers 结构开启 virustotal 功能。之后蜜罐如果成功下载恶意文件，则会自动提交到 virustotal，提交信息和反馈的检测信息保存到数据库中。

virustotalscan	virustotal	virustotalscan_scanner	virustotalscan_result
1	1	nProtect	Worm/W32.Kido.162516
2	1	CAT-QuickHeal	I-Worm.Kido.dt
3	1	McAfee	Artemis!D0E0C049ED70
4	1	TheHacker	W32/Kido.dt
5	1	K7AntiVirus	NetWorm
6	1	VirusBuster	Worm.Kido!I5VsGjjz6XY
7	1	NOD32	Win32/Conficker.AE
8	1	F-Prot	W32/Conficker!Generic
9	1	Symantec	W32.Downadup.B
10	1	Norman	W32/Conficker.KJ
11	1	TrendMicro-HouseCall	WORM_DOWNAD.AD
12	1	Avast	Win32:Malware-gen

图5 Virustotal 扫描反馈的部分结果（Conficker 蠕虫）

5.4. XMPP 日志提交

Dionaea 支持分布式部署，蜜罐节点可以通过 xmpp 协议将捕获的攻击的信息（与事件数据库文件 logsql.sqlite 中内容相同）和下载的恶意文件提交到中心服务器。使用此功能需在 Dionaea 配置文件中 ihandlers 结构开启 xmpp 提交功能，在 xmpp 提交配置选项部分填写提交参数，同时在中心服务器安装 xmpp 服务程序，中心数据库，并运行将 xmpp 服务收到的数据记录到数据库中的脚本。

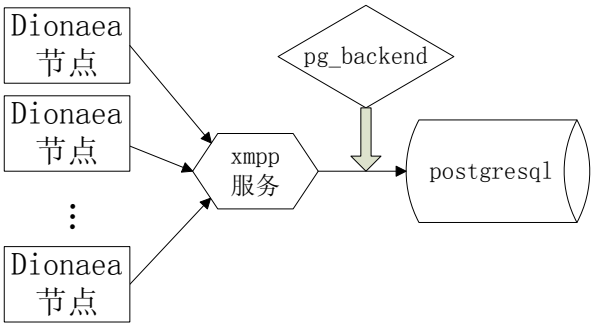

图 6 xmpp 日志提交流程

如图，每个 dionaea 节点将捕获的信息发送到 xmpp 服务器，dionaea 源码包中提供的 python 脚本 modules/pythom/util/xmpp/pg_backend.py 可以提取 xmpp 记录的信息，并将这些信息储存到数据库中。XMPP 服务和数据库安装在同一台主机，数据库默认使用 postgresql，安装好后用源码包中 modules/pythom/util/xmpp/pg_schema.sql 文件完成建表操作。这样，分布式蜜罐捕获的信息集中到 postgresql 数据库中，可以通过 pgadmin 等管理工具或 web 方式查看。

Dionaea 利用 XMPP 日志提交实现分布式功能，通过查询中心数据库，可以方便快捷地获取大范围内网络安全威胁信息，为进一步分析提供基本条件。

6. 总结

Dionaea 低交互式蜜罐是一款安装配置简单，具有较强恶意攻击捕获能力和扩展能力的蜜罐系统，它支持分布式部署，可以很方便地通过大范围部署提高对网络恶意攻击的监测能

力。当然，Dionaea 不能完全避免低交互式蜜罐的普遍弱点，即对网络服务的模拟与真实服务存在差距，可能无法捕获某些对环境敏感的攻击，这个缺点目前只能通过不断扩展模拟服务脚本，以及结合专用服务的蜜罐软件来进行完善。

在下篇中，我们将对在清华校内网络和 CERNET 骨干网上实际部署的两个 Dionaea 蜜罐实例所捕获的数据进行分析，一方面来了解目前教育网中的网络服务端威胁现状，另一方面也将展示 Dionaea 的威胁捕获实际能力。