Zeek部署使用手册

****

|  |  |  |  |
| --- | --- | --- | --- |
| 文件类型 | 文献翻译/使用手册/分析报告 | 密 级 | 商业秘密 |
| 编制人 | 听风者实验室 | 版 本 | V\*.\* |
| 审 核 |  | 审核日期 |  |
| 批准人 |  | 批准日期 |  |

**北京天融信网络安全技术有限公司**

**201X年XX月XX日**

**文档修订**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **日期** | **修改人员** | **描述** | **审核人员** |
| V1.0 | 2019.04.03 | 秦纪伟 | 创建初稿 |  |
|  |  |  |  |  |

**版权说明**

本文件中出现的全部内容，除另有特别注明，版权均属北京天融信网络安全技术有限公司所有。任何个人、机构未经北京天融信网络安全技术有限公司书面授权许可，不得以任何方式复制或引用文件的任何片断。

**保密申明**

本文件包含了来自北京天融信网络安全技术有限公司听风者实验室可靠、权威的信息，以及被检测单位信息系统的敏感信息，接受这份文件表示同意对其内容保密并且未经北京天融信网络安全技术有限公司书面请求和书面认可，不得复制，泄露或散布这份文件。如果你不是有意接受者，请注意对这份文件内容的任何形式的泄露、复制或散布均被禁止。

**完成报告后删除红色部分**

报告格式内容要求：

1. 正文字体“宋体--小四”，1.5倍行距；表格字体用“宋体--五号”，行距从美观的角度自行把握，可设置为16-18磅
2. 目录行距设置为1.5倍
3. 表格行数较多时添加“序号”列，内容居中
4. **贴图务必使用自己的截图，杜绝使用友商的截图！**
5. 文章完成后检查一遍，注意不要有错别字

样本分析报告注意：

1. 涉及客户敏感信息（名称、域名、IP、服务器主机名等）的文字，仅限出现在第一章“概述”中；分析过程中图片含有敏感信息的需打码
2. 添加“防护措施”章节，不少于4条，总共不少于100字

目 录

[Zeek部署使用手册 1](#_Toc23510555)

[1. 概述 1](#_Toc23510556)

[2. Zeek介绍 2](#_Toc23510557)

[2.1. Zeek历史 2](#_Toc23510558)

[2.2. Zeek功能 3](#_Toc23510559)

[2.2.1. 部署方式 3](#_Toc23510560)

[2.2.2. 分析框架 3](#_Toc23510561)

[2.2.3. 脚本支持 4](#_Toc23510562)

[2.2.4. 用户界面 4](#_Toc23510563)

[2.3. Zeek架构 4](#_Toc23510564)

[2.3.1. Zeek集群架构 5](#_Toc23510565)

[3. Zeek安装 6](#_Toc23510566)

[3.1. 安装依赖程序 6](#_Toc23510567)

[3.2. 源码安装依赖程序 6](#_Toc23510568)

[3.2.1. 可选的依赖程序 7](#_Toc23510569)

[3.2.2. 批量安装必要依赖程序 7](#_Toc23510570)

[3.3. 获得Zeek安装包 8](#_Toc23510571)

[3.3.1. Git下载 8](#_Toc23510572)

[3.3.2. 官网下载 8](#_Toc23510573)

[3.4. 源码安装Zeek 8](#_Toc23510574)

[3.4.1. 解压源码包 9](#_Toc23510575)

[3.4.2. 执行配置 9](#_Toc23510576)

[3.4.3. Make编译 12](#_Toc23510577)

[3.4.4. 配置环境变量 12](#_Toc23510578)

[3.4.5. 配置Zeek 13](#_Toc23510579)

[3.4.6. 启动Zeek 14](#_Toc23510580)

[4. 使用Zeek分析 14](#_Toc23510581)

[4.1. 监控实时流量 14](#_Toc23510582)

[4.2. 读取捕获包文件（PCAP） 14](#_Toc23510583)

[4.3. 使用zeek-cut工具 15](#_Toc23510584)

[4.4. Zeek日志文件 15](#_Toc23510585)

[4.4.1. 网络协议日志 15](#_Toc23510586)

[4.4.2. 文件传输日志 16](#_Toc23510587)

[4.4.3. 网络控制日志 17](#_Toc23510588)

[4.4.4. 网络检测日志 17](#_Toc23510589)

[4.4.5. 网络观察日志 17](#_Toc23510590)

[4.4.6. 解析异常日志 18](#_Toc23510591)

[4.4.7. Zeek诊断日志 18](#_Toc23510592)

[5. Zeek情报框架 19](#_Toc23510593)

[5.1. 使用方法 19](#_Toc23510594)

# 概述

无论是操作系统、应用软件、网络设备还是业务系统都普遍存在未知的漏洞，在网络攻击组织化的背景下，网络安全面临更加严峻的挑战。传统的安全检测方案基本都是基于已知的规则库进行监测，可以检测出已知威胁，对于未知威胁无法发挥更高的作用，并且对正在发生的或者已经造成损失的网络攻击行动无法做到完整的溯源取证和损失的评估。为了解决这个问题网络流量分析就成了有效的解决方案，因为不论什么样的网络攻击都会在网络中留下一些痕迹，通过分析网络流量来对网络攻击的行为进行识别和提供有效的解决方案。

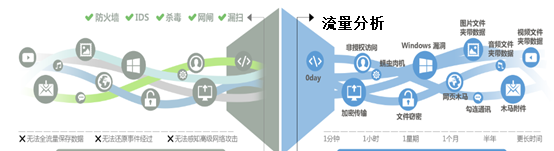


图:传统的网络安全检测的不足

在网络流量分析中对威胁深度识别、流量协议解析、应用识别、行为溯源以及流量趋势分析，再结合离线高可信的威胁情报库与风险检测模型实时掌握流量信息资产风险、敏感信息行为访问、僵尸网络和失陷主机等用户关注的流量行为。



图：网络流量威胁分析

# Zeek介绍

Zeek是一个被动的开源网络流量分析器，它主要是一种安全监视器可以深入检查链路上的所有流量分析可以的活动迹象，Zeek除了在安全领域的应用之外还支持各种流量分析任务包括网络性能监测以协议故障的排除。

在网络中部署的Zeek可以获得直接的成果就是处理好的大量日志文件，这些日志文件以协议分类记录了网络的活动。这些日志不仅包括对网络上每个连接的全面记录，还包括应用程序记录，例如所有HTTP会话和请求的URI、密钥、请求头、MIME类型和服务器的响应；带有回复的DNS请求；SSL证书；SMTP会话的相关内容等等。

除了日志Zeek还具有内置的一系列分析和监测任务的功能，包括从HTTP会话中提取文件，通过与外部注册表来检测恶意软件，检测网络上可见的易受攻击的软件版本，识别流量的网络、应用程序、检测SSH暴力破解、验证SSL证书链等。

## Zeek历史

Zeek在之前为”Bro”Zeek 的历史可以追溯到二十多年前由范恩·帕克森（Vern Paxson）设计并创建了最初的版本。范恩(Vern)于1995年在劳伦斯伯克利国家实验室(LBNL)担任研究员，负责编写Bro的代码。伯克利实验室于1996年开始进行业务部署，而USENIX安全研讨会在1998年发布了最初的Bro论文并在随后的期刊中进行了完善。2003年美国国家科学基金(NSF)会开始在国际计算机科学研究所(ICSI)上支持Bro的研究和高级开发，范恩现在领导该网络安全小组。多年来不断增长的国际计算机科学研究所的研究人员和学生团队不断增加新功能而伯克利国家实验室在能源部(DOE)的资助下继续也继续提供支持。

Zeek的大部分功能来自学术研究项目，其结果通常在顶级会议上发布。随着Zeek运营的用户社区的不断发展，以研究为中心的开发模型最终称为系统发展的瓶颈。NSF在2010年着手解决这个问题，通过向ICSI授予SDCI计划中专门用于Zeek开发的捐款，有了这个支持美国国家超级计算应用中心（NCSA）作为团队的核心合作伙伴也加入了该项目，并且该项目开始为2.0版本完全检查系统的许多用户可见部分。自从发布该版本以后，Zeek在各种新的部署中经历了巨大的增长，并且Zeek团队现在正在努力通过进一步提高系统的功能来应对未来网络的挑战。

## Zeek功能

### 部署方式

* 在标准UNIX风格的系统（包括Linux，FreeBSD和MacOS）上的商用硬件上运行。
* 通过网络分接头或监视端口进行完全被动的流量分析。
* 用于捕获数据包的标准libpcap接口。
* 实时和离线分析。
* 集群支持大规模部署。
* 统一管理框架，可用于独立和群集设置。
* 在BSD许可下开源。

### 分析框架

* 全面记录活动，以进行离线分析和取证。
* 与应用程序层协议无关的端口分析。
* 支持许多应用程序层协议（包括DNS，FTP，HTTP，IRC，SMTP，SSH，SSL）。
* 分析通过应用层协议交换的文件内容，包括用于指纹识别的MD5 / SHA1计算。
* 全面的IPv6支持。
* 隧道检测和分析（包括Ayiya，Teredo，GTPv1）。Zeek将隧道解封装，然后继续分析其内容，就好像没有适当的隧道一样。
* 在协议分析过程中进行全面的完整性检查。
* 支持IDS样式的模式匹配。

### 脚本支持

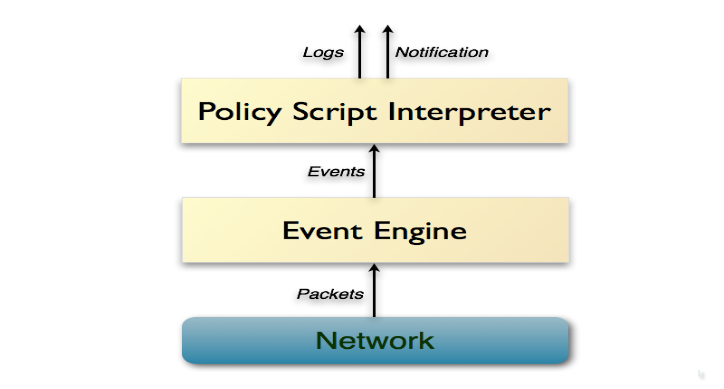
* 图灵完备的语言，用于表达任意分析任务。
* 基于事件的编程模型。
* 特定于域的数据类型，例如IP地址（透明地处理IPv4和IPv6），端口号和计时器。
* 长期支持跟踪和管理网络状态。

### 用户界面

* 默认输出为结构良好的ASCII日志。
* ElasticSearch和DataSeries的备用后端。正在准备其他数据库接口。
* 外部输入实时集成到分析中。正在准备实时数据库输入。
* 外部C库，用于与外部程序交换Zeek事件。带有Perl，Python和Ruby绑定。
* 能够从脚本语言内部触发任意外部进程。

## Zeek架构

Zeek的架构主要分为两个主要部分，它的事件引擎（或核心）将传入的数据包解析成一系列更高级别的事件，这些事件和策略无关的属于反映了网络活动，例如网络上的每个HTTP请求都变成了一个相应的http\_reauest事件带有相应的IP地址和端口，所请求的URI以及所使用的HTTP版本。

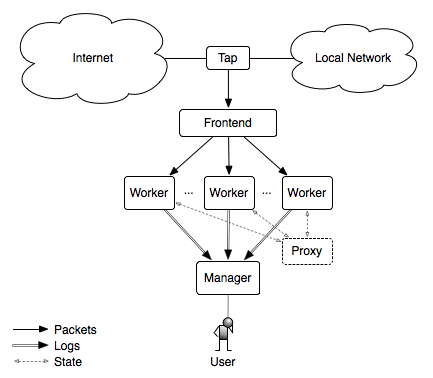


图：Zeek的内部架构

Zeek的第二个主要组件时脚本解析器，脚本解析器执行一组事件处理程序用Zeek的自定义脚本语言编写。这些脚本可以表达网络的的安全策略，当Zeek检测到不同类型的活动时要采取的操作。通常他们可以输入流量中得出任何所需的属性和统计信息。Zeek的语言具有广泛的特定于域的类型和支持功能。Zeek脚本可以生成实时告警，还可以按需执行任意外部程序，例如对攻击的主动响应。

### Zeek集群架构

Zeek不是多线程运行的程序，一旦达到单个处理器的内核限制，当前唯一的选择就是将负载分散到许多的内核甚至许多的物理计算机上面。Zeek的集群部署方案是构建较大网络环境的解决方案。Zeek附带的工具和脚本提供了一种结构，可以轻松的管理多个Zeek进程，以检查数据包并进行关联活动。



图：Zeek集群架构

Manager：Manager是一个Zeek流程，有两个主要任务。它使用Zeek通信协议从群集的其余节点接收日志消息和通知（请注意，如果使用单独的记录器节点，则记录器将接收所有日志，而不是管理器）。

Logger：Logger是一个可选的Zeek进程，它使用Zeek通信协议从群集中其余节点接收日志消息。让记录器代替管理器接收日志的目的是减轻管理器的负担。

Proxy：Proxy是Zeek进程，可用于卸载数据存储或任何任意工作负载。一个群集可能包含多个代理节点。Zeek随附的默认脚本使用的代理最少，因此仅使用一个代理就足够了。

Worker: 是Zeek进程，它嗅探网络流量并对重组后的流量进行协议分析。活动集群的大部分工作都在工作程序上进行，因此，工作程序通常代表集群中正在运行的大部分Zeek进程。

# Zeek安装

Zeek支持在标准的UNIX风格系统（包括Linux，FreeBSD和MacOS）上运行。Zeek可以使用预编译二进制包安装、源码安装两种方式。

## 安装依赖程序

在安装Zeek之前，需要确保已安装一些依赖项，Zeek需要在开始之前安装以下库和工具：

* Libpcap ([http://www.tcpdump.org](http://www.tcpdump.org/))
* OpenSSL libraries ([http://www.openssl.org](http://www.openssl.org/))
* BIND8 library
* Libz
* Bash (for ZeekControl)
* Python 2.6 or greater (for ZeekControl)

## 源码安装依赖程序

若要从源代码构建Zeek，还需要以下附加依赖项：

* CMake 2.8.12 or greater ([http://www.cmake.org](http://www.cmake.org/))
* Make
* C/C++ compiler with C++11 support (GCC 4.8+ or Clang 3.3+)
* SWIG ([http://www.swig.org](http://www.swig.org/))
* Bison 2.5 or greater (<https://www.gnu.org/software/bison/>)
* Flex (lexical analyzer generator) (<https://github.com/westes/flex>)
* Libpcap headers ([http://www.tcpdump.org](http://www.tcpdump.org/))
* OpenSSL headers ([http://www.openssl.org](http://www.openssl.org/))
* zlib headers (<https://zlib.net/>)
* Python (<https://www.python.org/>)

### 可选的依赖程序

* libmaxminddb（用于IP地址定位）
* sendmail（启用Zeek和ZeekControl发送邮件）
* curl（由实现活动HTTP的Zeek脚本使用）
* gperftools（tcmalloc用于提高内存和CPU使用率）
* jemalloc（<http://www.canonware.com/jemalloc/>）
* PF\_RING（仅Linux，用于集群架构）
* krb5库和头文件
* ipsumdump（用于跟踪摘要；[http：//www.cs.ucla.edu/~kohler/ipsumdump](http://www.cs.ucla.edu/~kohler/ipsumdump)）

### 批量安装必要依赖程序

#### 基于RPM / RedHat的Linux

sudo yum install cmake make gcc gcc-c++ flex bison libpcap-devel openssl-devel python-devel swig zlib-devel

#### 基于DEB / Debian的Linux

sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libssl-dev python-dev swig zlib1g-dev

注：如果使用Python 2.7，则还需要安装“ python-ipaddress”软件包。

#### FreeBSD

sudo pkg install bash cmake swig30 bison python py27-sqlite3 py27-ipaddress

#### Mac OS X

在Mac上编译源代码需要先安装Xcode 或“命令行工具”（下载量要小得多）。要检查是否已安装，请运行命令xcode-select --install。

## 获得Zeek安装包

可以通过Zeek官网下载源码压缩包或者使用Git下载git存储库中的源码包。

### Git下载

git clone --recursive https://github.com/zeek/zeek

### 官网下载

首先检查OpenPGP签名 。见新闻发布说明和变化 的相比过去的版本变化的完整列表。

下载链接：<https://www.zeek.org/download/index.html>



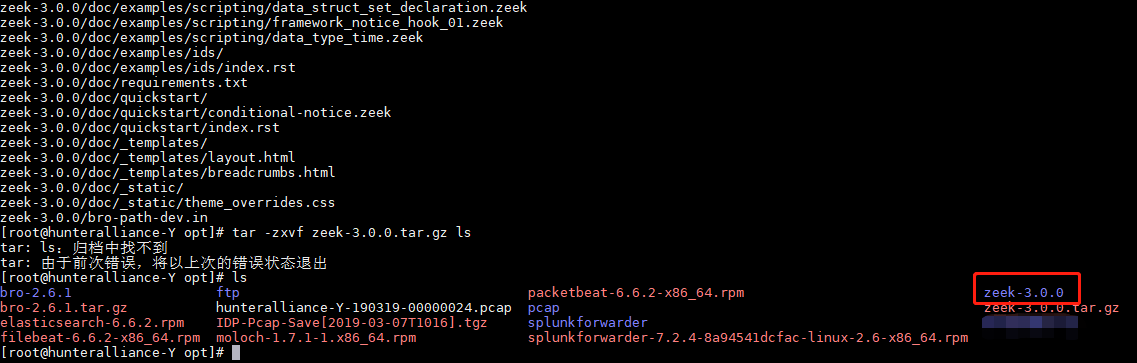
图：Zeek官网下载页面

## 源码安装Zeek

从官方网站进行源码包的下载后上传到服务器的目录下进行解压并配置安装，当前步骤默认已完成依赖环境的配置后进行Zeek的安装步骤。

### 解压源码包

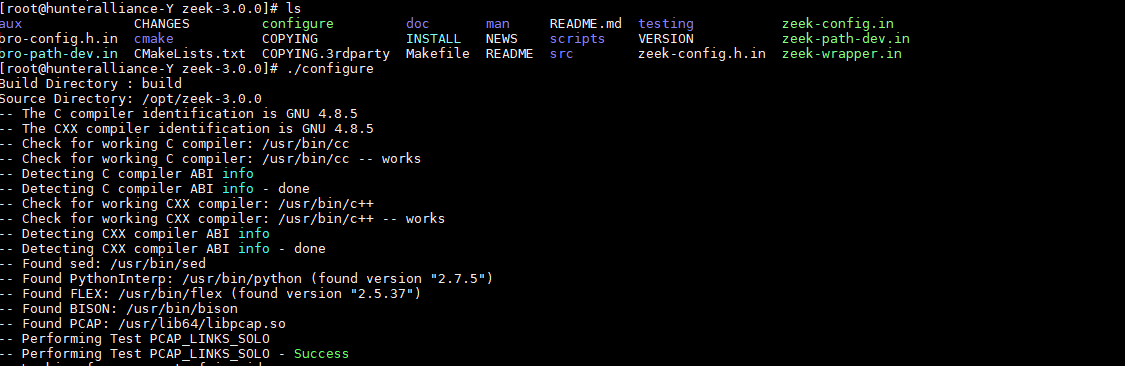
[root@hunteralliance-Y opt]# tar -zxvf zeek-3.0.0.tar.gz



图：Zeek解压完毕

### 执行配置

[root@hunteralliance-Y zeek-3.0.0]# ./configure



图：执行Zeek默认配置

此处执行configure 采用的时默认配置，如果要进行更多的配置可使用 ./configure –-help来获取更多参数。

用法：./ configure [OPTION] ... [VAR = VALUE] ...

构建选项：

--builddir =目录[build]中的DIR位置构建文件

--build-type = TYPE set CMake构建类型[RelWithDebInfo]：

调试：优化关闭，调试符号+标志

MinSizeRel：大小优化，调试关闭

发布：优化，调试关闭

RelWithDebInfo：优化，

调试符号打开，调试标志关闭

--generator = GENERATOR CMake生成器使用（参见cmake --help）

--ccache使用ccache来加速重新编译（需要

ccache安装和CMake 3.10+）

--toolchain = CMAKE\_TOOLCHAIN\_FILE的PATH路径

（对于交叉编译很有用）

安装目录：

--prefix = PREFIX安装目录[/ usr / local / bro]

--scriptdir = Bro脚本的PATH根安装目录

[PREFIX /共享/ BRO]

--localstatedir = PATH使用BroControl时，存储日志文件的路径

和运行时数据（在log /和spool / subdirs中）

[字首]

--spooldir = PATH使用BroControl时，存储运行时数据的路径

[PREFIX /阀芯]

--logdir = PATH使用BroControl时，存储日志文件的路径

[PREFIX /日志]

--conf-files-dir = PATH配置文件安装目录[PREFIX / etc]

可选功能：

--enable-debug在调试模式下编译（如--build-type = Debug）

- 支持代码覆盖的--enable-coverage编译（暗示调试模式）

--enable-mobile-ipv6分析RFC 6275定义的移动IPv6功能

--enable-perftools强制在非Linux系统上使用Google perftools

（在Linux上存在perftools时自动打开）

--enable-perftools-debug使用Google的perftools进行调试

--enable-jemalloc链接反对jemalloc

--enable-broccoli构建或安装Broccoli库（不建议使用）

静态--enable-static-broker构建代理（如果指定了--with-broker，则忽略）

--disable-broctl不安装Broctl

--disable-auxtools不构建或安装辅助工具

--disable-perftools不会尝试使用Google Perftools构建

--disable-python不会尝试为代理构建python绑定

--disable-broker-tests不会尝试构建Broker单元测试

非标准位置中的必需包：

--with-openssl = OpenSSL安装根的PATH路径

--with-bind = BIND安装根的PATH路径

--with-pcap = libpcap安装根的PATH路径

--with-binpac = BinPAC可执行文件的PATH路径

（对于交叉编译很有用）

--with-bifcl = Bro BIF编译器可执行文件的PATH路径

（对于交叉编译很有用）

--with-flex = flex可执行文件的路径

--with-bison =野牛可执行文件的路径

--with-python = Python可执行文件的PATH路径

--with-broker = Broker安装root的PATH路径

（Bro默认使用嵌入式版本）

--with-caf = PATH路径到C ++ Actor Framework安装root

（默认情况下嵌入的Broker依赖项）

非标准位置中的可选包：

--with-geoip = libmaxminddb安装根目录的PATH路径

--with-krb5 = krb5安装root的PATH路径

--with-perftools = Google Perftools安装root的PATH路径

--with-jemalloc = jemalloc安装根的路径

--with-python-lib = libpython的PATH路径

--with-python-inc = Python头的PATH路径

--with-swig = SWIG可执行文件的PATH路径

--with-rocksdb = RocksDB安装的路径路径

（可选的Broker依赖）

包装选项（适用于开发人员）：

--binary-package toggle二进制打包的特殊逻辑

--ignore-dirs =创建源包时要忽略的PATHS路径

（分号分隔并在多个时引用）

--pkg-name-prefix = NAME使用给定的名称作为包前缀

默认的CMake项目名称

--osx-sysroot =要编译的OS X SDK的PATH路径

--osx-min-version = VER最小OS X版本（部署目标）

有影响力的环境变量（仅在第一次调用时

每个构建目录）：

CC C编译器命令

CFLAGS C编译器标志

CXX C ++编译器命令

CXXFLAGS C ++编译器标志

### Make编译

[root@hunteralliance-Y zeek-3.0.0]# make && make install

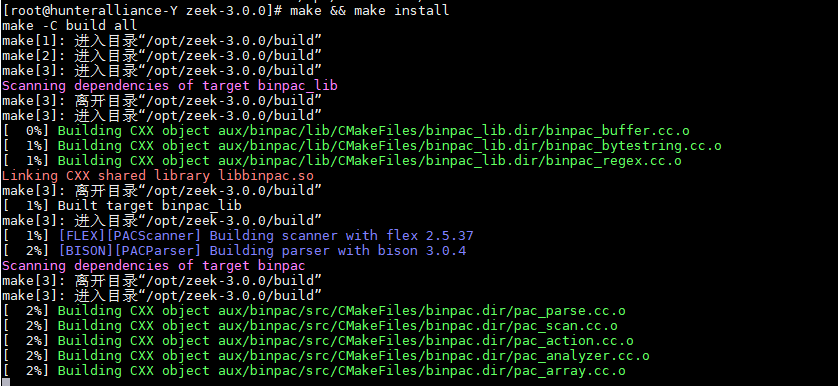


图:编译Zeek源码并安装

### 配置环境变量

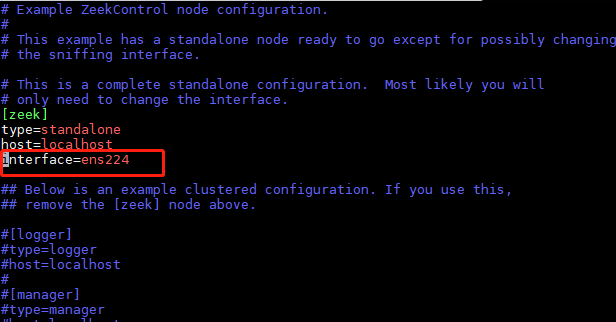
[root@hunteralliance-Y zeek-3.0.0]# export PATH=/usr/local/zeek/bin:$PATH

### 配置Zeek

在完成Zeek的编译安装之后需要对配置文件进行配置后才可以启动。

#### 配置网卡监听

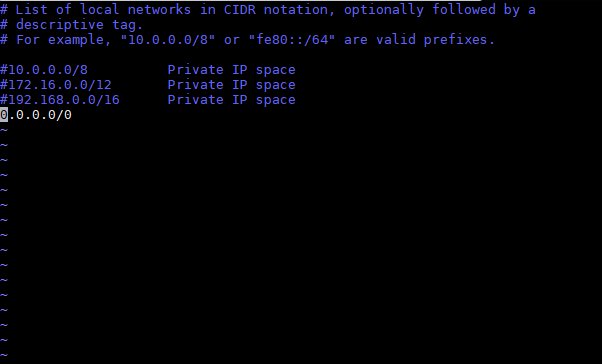
编辑$etc/nodes.cfg



图：配置Zeek网卡监听

#### 配置网络监控范围

编辑$etc/network.cfg,将默认的配置注释后添加任意网络范围。



图：配置Zeek网络监听范围

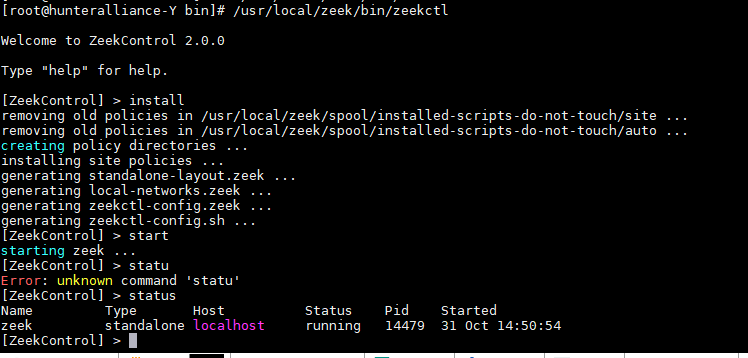
### 启动Zeek

[root@hunteralliance-Y bin]# /usr/local/zeek/bin/zeekctl

[ZeekControl] > install

[ZeekControl] > start

[ZeekControl] > status



图：启动Zeek

# 使用Zeek分析

## 监控实时流量

[root@hunteralliance-Y bin]# ./zeek -i ens224

## 读取捕获包文件（PCAP）

[root@hunteralliance-Ytmp]#/usr/local/zeek/bin/zeek-r f490dea01cdb0a8ef3d7f4802f78e0f3.pcap



图：Zeek读取PCAP流量包

## 使用zeek-cut工具

Zeek-cut可以使用该程序来代替其他工具来执行终端命令，这些命令可以独立于日志文件本身的更改而保持灵活而准确的状态。

查看指定字段内容：cat conn.log | zeek-cut id.orig\_h id.orig\_p id.resp\_h duration

解析时间戳：zeek-cut -d ts uid host uri < http.log

检测UID：cat conn.log | zeek-cut uid resp\_bytes | sort -nrk2 | head -5

## Zeek日志文件

Zeek的日志文件保存在$zeek/logs目录下面，Zeek默认情况下会按照每天新建一个以日期作为文件名，然后按照每小时创建一个压缩包存储在目录中。并且在$/zeek/logs/current目录中会将当前实时捕获到的记录按照协议类型保存在其中。以下对Zeek的日志记录进行介绍。

### 网络协议日志

表：Zeek网络协议日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | conn.log | TCP / UDP / ICMP连接 |
| 2 | dce\_rpc.log | 分布式计算环境/ RPC命令执行 |
| 3 | dhcp.log | DHCP租约 |
| 4 | dnp3.log | DNP3请求和答复 |
| 5 | dns.log | DNS活动 |
| 6 | ftp.log | FTP活动 |
| 7 | http.log | HTTP请求和回复 |
| 8 | irc.log | IRC命令和响应 |
| 9 | kerberos.log |  |
| 10 | modbus.log | Modbus命令和响应 |
| 11 | modbus\_register\_change.log | 跟踪对Modbus保持寄存器的更改 |
| 12 | mysql.log | Mysql协议命令执行 |
| 13 | ntlm.log | NT LAN管理器（NTLM） |
| 14 | radius.log | RADIUS身份验证尝试 |
| 15 | rdp.log | RDP活动 |
| 16 | rfb.log | 远程帧缓冲（RFB）远程帧缓冲 |
| 17 | sip.log |  |
| 18 | smb\_cmd.log | SMB命令执行 |
| 19 | smb\_files.log | SMB文件传输 |
| 20 | smb\_mapping.log | SMB树 |
| 21 | smtp.log | SMTP会话 |
| 22 | snmp.log | SNMP会话 |
| 23 | socks.log | Socks代理 |
| 24 | ssh.log | SSH会话 |
| 25 | ssl.log | SSL / TLS握手信息 |
| 26 | syslog.log | Syslog消息 |
| 27 | tunnel.log | 隧道协议事件 |

### 文件传输日志

表：Zeek文件传输日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | files.log | 文件分析结果 |
| 2 | ocsp.log | 在线证书状态协议（OCSP）。仅在加载策略脚本时创建。 |
| 3 | pe.log | 便携式可执行文件（PE） |
| 4 | x509.log | 加密协议证书信息 |

### 网络控制日志

表：Zeek网络控制日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | netcontrol.log | NetControl动作 |
| 2 | netcontrol\_drop.log | NetControl动作 |
| 3 | netcontrol\_shunt.log | NetControl分流动作 |
| 4 | netcontrol\_catch\_release.log | NetControl捕获和释放操作 |
| 5 | openflow.log | OpenFlow调试日志 |

### 网络检测日志

表：Zeek网络检测日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | intel.log | 情报数据匹配 |
| 2 | notice.log | Zeek默认告警 |
| 3 | notice\_alarm.log | 警报流 |
| 4 | signatures.log | 签名匹配 |
| 5 | traceroute.log | 跟踪路由检测 |

### 网络观察日志

表：Zeek网络观察日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | known\_certs.log | SSL证书 |
| 2 | known\_hosts.log | 完成TCP握手的主机 |
| 3 | known\_modbus.log | Modbus主站和从站 |
| 4 | known\_services.log | 在主机上运行的服务 |
| 5 | software.log | 网络上使用的软件 |

### 解析异常日志

表：Zeek解析异常日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | barnyard2.log | 从Barnyard2收到的警报 |
| 2 | dpd.log | 动态协议检测失败 |
| 3 | unified2.log | 解释Snort的统一输出 |
| 4 | weird.log | 意外的网络级活动 |
| 5 | weird\_stats.log | 有关意外活动的统计 |

### Zeek诊断日志

表：Zeek诊断日志

|  |  |  |
| --- | --- | --- |
| 序号 | 日志文件 | 描述 |
| 1 | broker.log | Zeek或启用Broker的流程之间的对等状态事件 |
| 2 | capture\_loss.log | Zeek捕获耗损 |
| 3 | cluster.log | Zeek集群消息 |
| 4 | config.log | 配置选项更改 |
| 5 | loaded\_scripts.log | 显示Zeek加载的所有脚本 |
| 6 | packet\_filter.log | 列出已应用的数据包筛选器 |
| 7 | prof.log | 分析统计信息（要创建此日志，请加载policy / misc / profiling.zeek |
| 8 | reporter.log | 内部错误/警告/信息消息 |
| 9 | stats.log | 内存/事件/数据包/滞​​后统计 |
| 10 | stderr.log | 从ZeekControl启动Zeek时捕获标准错误 |
| 11 | stdout.log | 从ZeekControl启动Zeek时捕获标准输出 |

# Zeek情报框架

威胁情报数据对于安全检测过程非常重要，通过事件响应过程中发现的数据和通过私有社区共享的数据。Zeek的情报框架就是使用这种数据，可以用于匹配并提供高性能和内存利用优化。情报框架中的数据基本组成部分IP地址和电子邮件，以及其他的元数据。

## 使用方法

redef Intel::read\_files += { "/somewhere/yourdata.txt" };