Ehoney欺骗防御系统

None

licheng

None

Table of contents

1. 我们的故事

e签宝是一个初创公司，比较穷，买不起各种安全产品，又面临各种安全问题，我们是依赖阿里云，但是用不起阿里云的安全产品，在结合我以前的工作经验，都喜欢自研安全产品，所以就努力说服老板招人自己做，老板被忽悠成功了，就招聘7个人左右，小公司没名气，给不起钱，招人是件痛苦的事情，我们光建立这个安全团队就花费了2年时间(不仅仅是钱的问题)，2年是730天.这里为什么强调天，因为可能在e签宝安全团队每天都是在战斗。说了这么多(废话比较多)，还没说到为啥要做开源，由于我们公司是做saas的，我们自己就爱买saas产品，我们会做安全评估，发现我们采购的所有产品，都存在很严重的安全，TOB这个行业本身就比较苦逼，更没有精力投入安全，他们也面临跟我们一样的问题，我们比他们好点的是有个明智的老板，所以团队小伙伴们就想着能不能把我们自研的安全产品开源出去，多么单纯的想法啊，做起来才发现TM都是坑！

开源啥东西，怎么开源，其实都不知道，只是一股热情，所以就想到开源老鸟，吴敏，吴博士，后续简称老吴，老吴在开源路上给了我们很大的帮助，这里必须感谢下老吴的无私奉献，革命友谊长存，一直欠老吴去一次花都，某天晚上全体开源小组人员开到老吴创业的办公室，听老吴讲如何开源，老吴拿出了讲了N次的PPT给我们讲了一遍.说来也奇怪，每次听完都有不一样的收获，归结原因可能是老吴每次忽悠的都不一样吧，老吴挺能说的，说了4个多小时，兄弟们听的津津有味，实在忍不住肚子饿，就拉着一票兄弟去吃饭，当时已经23点多了，基本饭店都关门了，最后找到一家火锅店，大家吃了起来，这时大家都信心满满，充满期待，不过就是没喝酒，不过这个晚上也确实不一样，因为Ehoney在这个地方梦想起飞！

回去就开始弄开源，这中间又出了一档子事情，信通院也在搞开源，也在做先进安全能力评估，我这个人就是既要又要还要，就让项目负责人徐吉人去申报信通院的评审，评审时间只有一个月，我们开源软件屁都没有呢，就要参加比赛，硬着头皮上，徐吉为了能够参加比赛，拉上项目组小伙伴郑有乐，开始疯狂加班干，人多有多大胆，地有多大产的精神发挥了，其实在参加比赛前几天这东西流程还没跑通，所以比赛就要延期参加，这个也是我们后续项目的延期的开始，由于参加比赛的都是商业产品，也让徐吉开了眼界，认识到我们产品的不足和改进。

比赛参加完以后，回来就是对系统的又一次大改，定了一个开源发布的时间，这个时间变成了一个遥不可及的时间，项目一直延期，每次的预演都变成一次重构，团队同学也是越做也没信心，连续几个月每天工作强度十几个小时，就是没有结果，光打鸣不下蛋，作为团队负责人的我也是十分焦虑，但是我一直没有动摇开源的信念和对团队的信心，最后我也是坐不住了，开始两条腿走路，找新同学高峰对架构进行重新开发，老版本则让徐吉继续优化，我也亲自上阵，抓项目，设计，架构，文档，体验，每周预演一次，每次都骂，最后骂的大家都生病了，这个项目徐吉流鼻血，郑有乐全身过敏，阿飞想跑路，作为老大的我，内心很无奈，也很感动，这更加坚定，要把东西做出来，做出价值。

今天是我们第一次提交代码，写这篇文章进行纪念，我相信这是一个伟大的产品的开始，"星星之火，可以燎原"，相信这个团队会给安全开源带来不一样的血液，如Indiana Jones，永远不会放弃找寻圣杯，我也不会放弃给自己热爱的事业工作，无论其他选择是如何如何的安全。当我老去甚至挂掉，我回顾我的一生，"wow，真是一个史诗般的冒险故事啊"而不会说"wow，我一生过得真是平稳无比"。最后也请使用我们产品的各位小伙伴，也请给这年轻的团队更多的包容和鼓励！

最后更新: 2021年6月28日

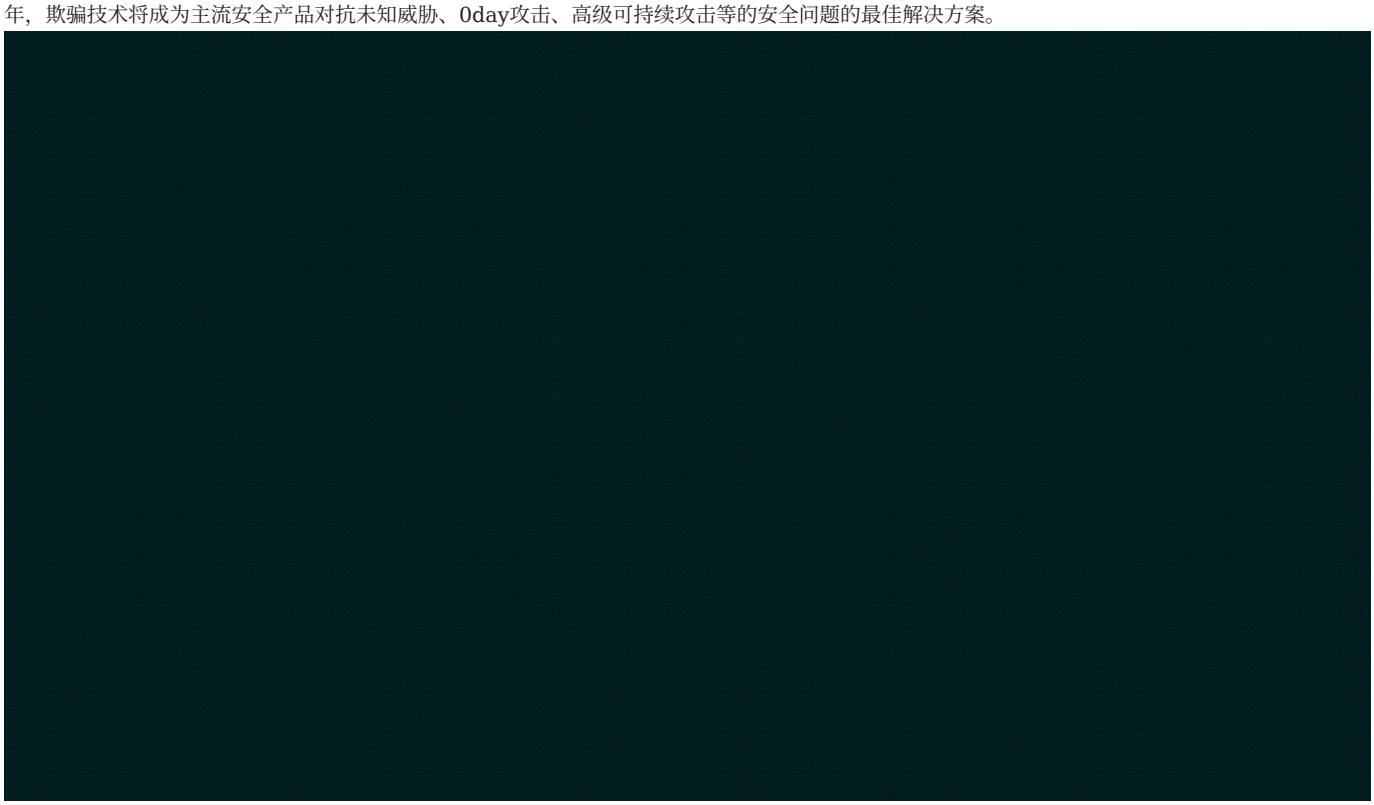
2. 介绍

2.1 什么是Ehoney

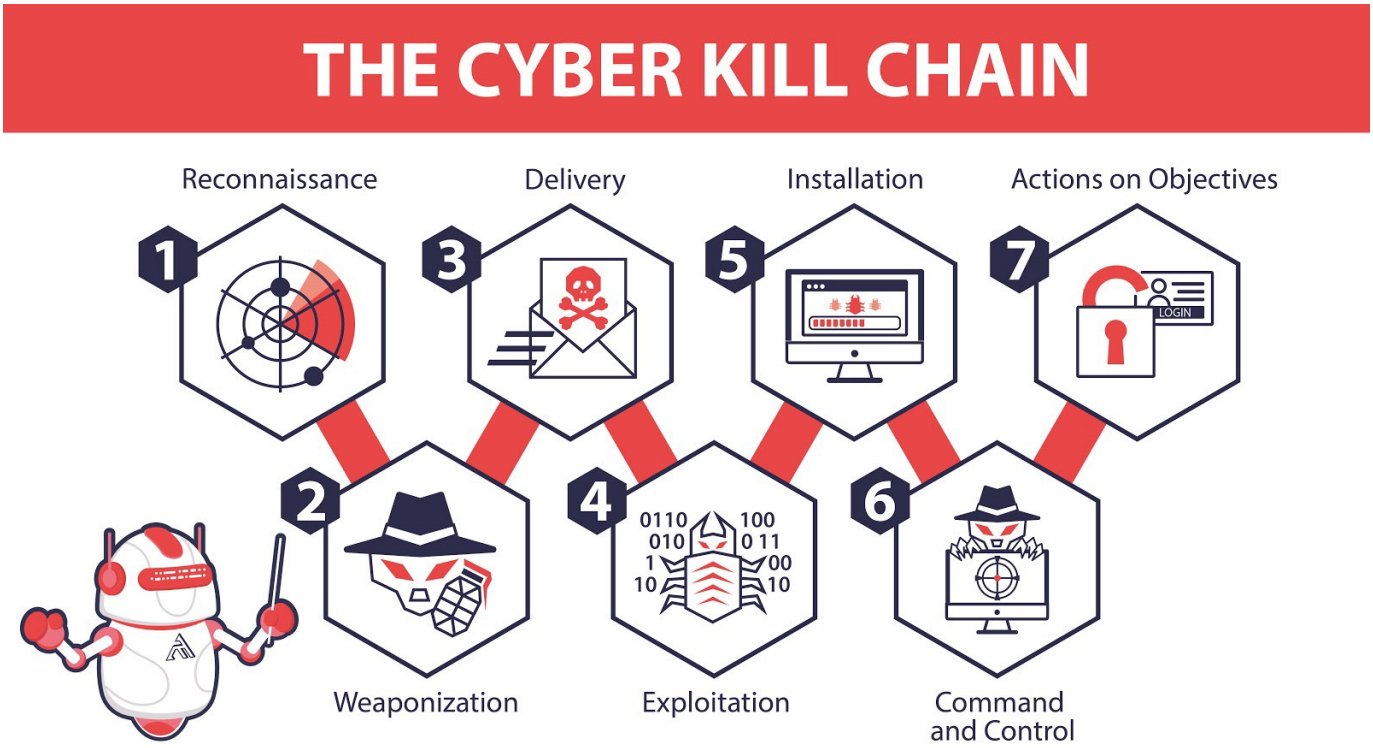
e签宝安全团队积累十几年的安全经验，都将对外逐步开放，首开的Ehoney欺骗防御系统，该系统是基于云原生的欺骗防御系统，也是业界唯一开源的对标商业系统的产品，欺骗防御系统通过部署高交互高仿真蜜罐及流量代理转发，再结合自研密签及诱饵，将攻击者攻击引导到蜜罐中达到扰乱引导以及延迟攻击的效果，可以很大程度上保护业务的安全。护网必备良药。

2.2 什么是欺骗防御

《孙子兵法》说，兵者，诡道也。从古至今，欺骗技术就作为战场上一种积极防御策略被一直使用。而网络欺骗技术，就是信息安全战场上防御者的“诡计”。Gartner在2016年安全与风险管理峰会的10大信息安全技术中，就提出了欺骗技术，Gartner对网络欺骗技术的定义为：使用骗局或者假动作来阻挠或者推翻攻击者的认知过程，扰乱攻击者的自动化工具，延迟或阻断攻击者的活动，通过使用虚假的响应、 有意的混淆、以及假动作、误导等伪造信息达到“欺骗”的目的。 欺骗技术(Deception Technology)已连续三年被Gartner列为十大安全技术之一， Gartner 认为，未来 5-10 年，欺骗技术将成为主流安全产品对抗未知威胁、0day攻击、高级可持续攻击等的安全问题的最佳解决方案。



2.3 解决的问题



`Cyber Kill Chain 不仅仅是一种攻击模型。Cyber Kill Chain 的七个阶段为任何组织的安全设计提供了极好的基础。`

- **侦察追踪** 攻击者收集有关其目标的信息。这包括间接和被动方法，攻击者通过这些方法从 ARIN（美国互联网号码注册机构）注册、Shodan 或工作列表等公共资源中收集信息。然后攻击者将转向更直接和主动的方法，例如端口扫描。
- **武器构建** 攻击者现在利用侦察中发现的漏洞发起攻击；许多利用来自 metasploit、exploit-db 或社会工程工具包的工具。
- **载荷投递** 一旦攻击者选择了最适合利用您的漏洞的工具，他们就会选择传送方法，无论是网络钓鱼电子邮件、受感染的 USB 还是其他选择的方法。
- **漏洞利用** 武器已交付。攻击者只需要执行攻击，它可以采用 SQL 注入、缓冲区溢出、RCE 以及无数其他形式。
- **安装植入** 攻击者获得更好的访问权限。常见的对抗技术包括在无文件攻击中执行 powershell、安装远程访问工具 (RAT) 和 DLL 劫持。
- **持续控制** 攻击者现在设置对您的系统的持久访问以进行远程操作。根据他们的最终目标，攻击者可以立即采取行动，也可以潜伏在您的系统中数月到数年。即使在重新启动或修补初始漏洞后，访问仍可能持续存在，并且通常被加密和屏蔽以使其看起来像正常流量。有时，它甚至嵌入在正常的合法流量中，例如 Twitter 或电子邮件。
- **目标达成** 攻击者实现他们的入侵目标，例如数据泄露、数据破坏或拒绝服务。

`然而，正如网络安全中的一句老话，“防御者需要防御一切，而攻击者只需要利用一个弱点。”，师以长夷以制夷，我们也可以用黑客的手段，反制黑客，打造一套覆盖整个攻击链的欺骗体系`

2.4 与蜜罐的不同：

- **仿真环境** 也就是大家理解的蜜罐，现在市场主流的蜜罐分高中低三类，诉求和解决的问题也不一样，如果要真实模拟环境，高交互蜜罐最合适，实现起来也更复杂，但是市面大部分蜜罐都低交互，稍微有点经验的黑客就能识破，何来欺骗！
- **覆盖率** 蜜罐是被动放在那里，等待黑客自己进来，比如线上服务器1w台，你不可能去部署1w台蜜罐，如果蜜罐仅仅部署几台，犹如杯水车薪，防御效果可想而知，所以这个是欺骗防御必须要解决的问题，就是如何做到高效的请君入瓮
- **攻击溯源** 如果采用高交互蜜罐，黑客入侵进去以后，怎么记录所有黑客的攻击，在蜜罐里装监控，黑客很容易就能发现，而且还能kill该监控，一般黑客都是攻击脚本不落盘，木马程序直接内存运行，没有办法拿到黑客样本，溯源非常困难
- **动态对抗** 蜜罐仅仅只能对攻击进行溯源，分析，不能做到根据黑客的行为，预测黑客的下一步，做到防范于未然，这个不仅仅是分析能力不足，蜜罐的架构也是没法实现动态对抗
- **安全风险** 黑客入侵蜜罐，如果蜜罐没做任何网络隔离，可能就会通过蜜罐做横向渗透测试

2.5 e签宝欺骗防御技术点

- **云原生** 轻量级k3s，满足容器管理，这样在一台机器上，可以构造无限制(受限机器配置)的仿真环境，有云原生的一切能力，而且还可以应用云原生自身的安全能力可以监控容器，如falco监控容器并且捕获攻击脚本，k3s防火墙限制只有容器之间互通，跟外界网络都是不通，解决了安全风险问题
- **代理技术** 代理分为透明代理，协议代理，透明代理部署在业务服务器上，解决了蜜罐覆盖率不够的问题，这样业务服务器仅仅安装透明代理，就可以把攻击流量牵引到蜜罐中，协议代理是建立在透明代理和k3s容器之间的桥梁，所有流量都通过协议代理转到对应的k3s容器中，这样既可以记录所有攻击请求，又可以让黑客毫无感知
- **诱饵(honeybit)** 黑客进入蜜罐，一般会做横向渗透测试，上图供给链也已经描述，比如黑客会查看history等，诱饵就是在history等地方插入蜜罐的信息，这样黑客攻击又会进入下一个蜜罐，环环相扣，仿佛进入迷宫一样
- **密签(honeytoken)** 黑客入侵主要还是拿到你的数据，但是大部分不知道数据是否泄露，如果黑客拿到数据，一打开数据就能定位黑客在哪，他是谁，是不是就能闭环，所以密签是整个欺骗防护的精华，所以其实密签可以是单独一个数据泄露产品，e签宝安全团队后续准备在单独开放出来
- **机器学习** 此处暂不做过多赘述，这是实现动态对抗蜜罐的核心，技术还在内部测试当中

2.6 e签宝欺骗防御系统的特性：

- **支持丰富的蜜罐类型**

  通用蜜罐： SSH 蜜罐、Http蜜罐、Redis蜜罐、Telnet蜜罐、Mysql蜜罐、RDP 蜜罐 IOT蜜罐： RTSP 蜜罐 工控蜜罐： ModBus 蜜罐

- **基于云原生技术**

  基于k3s打造saas平台欺骗防御，无限生成蜜罐，真实仿真业务环境

- **业内独一无二密签技术**

  独创的密签技术，支持20多种密签，如文件、图片，邮件等

- **强大诱饵**

  支持数十种诱饵，通过探针管理，进行欺骗引流

- **可视化拓扑**

  可以可视化展示攻击视图，让所有攻击可视化，形成完整的攻击链路

- **动态对抗技术**

  基于LSTM的预测算法，可以预测黑客下一步攻击手段，动态欺骗，延缓黑客攻击时间，保护真实业务

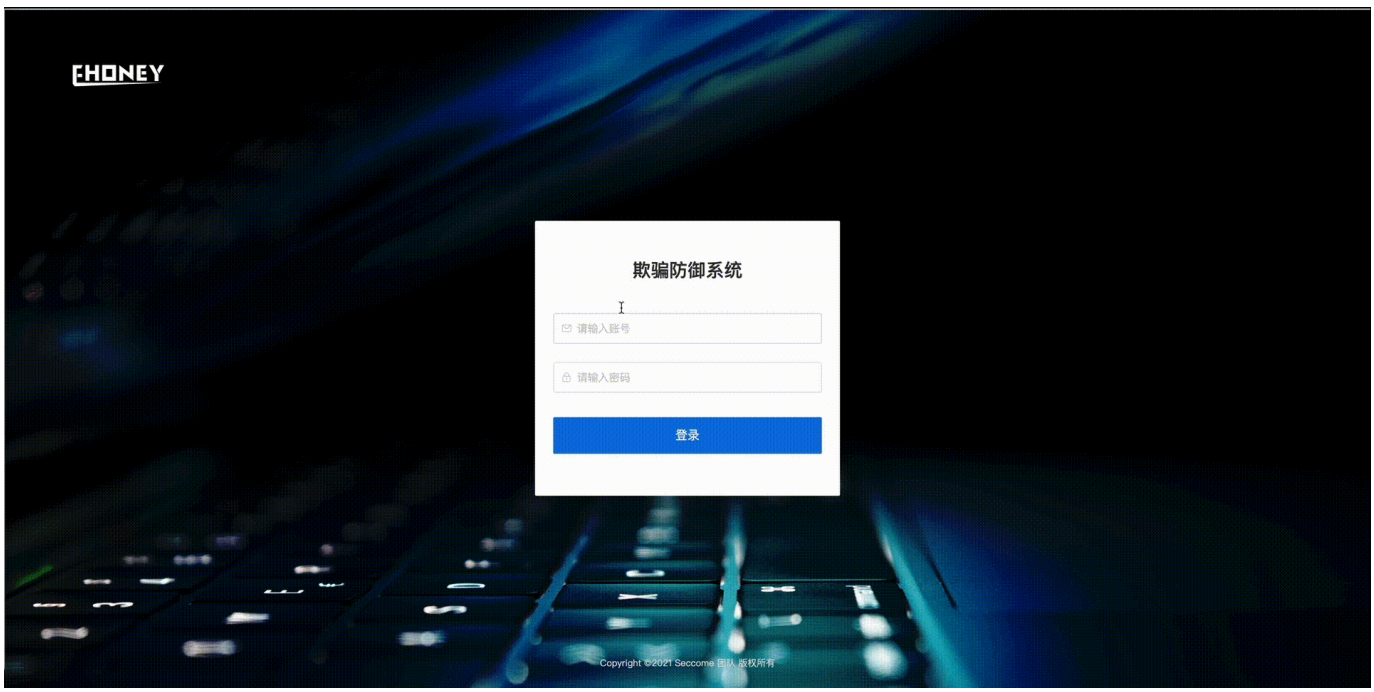
- **强大的定制化**

  支持自定义密签、诱饵、蜜罐等，插件化安装部署，满足一切特性需求

2.7 环境准备

- **系统要求**: CentOS 7 及以上
- **最低配置：**: 内存4G、磁盘空间10G以上
- **建议配置：**: 内存8G、磁盘空间30G以上

2.8 演示视频



2.9 相关链接

官网 文档首页 github

最后更新: 2021年6月29日

2.10 名词解释

蜜罐

- 容器化部署，全平台高交互蜜罐
- 仿真业务、协议等，吸引黑客攻击，
- 对攻击行为进行捕获和分析，了解攻击方所使用的工具与方法
- 推测攻击意图和动机

蜜网

- 是一种新型的蜜罐系统架构，通过将蜜罐集中的部署在一个受控的蜜网中，统一的进行数据分析和产生告警，而在真实的生产网络区域中仅仅部署一些轻量级的重定向模块，探针会通过监听相应的端口，将自己伪装成一个蜜罐，当探针受到攻击后会透明的将攻击流量转发到蜜场的蜜罐中。

探针

- 应用服务器，一般为黑客的第一道攻击入口

密签

- 定位攻击者 数据、文件、程序里加入特定标记，识别数据泄露，支持文件密签、邮件密签、图片密签、DNS密签、git密签、sql密签等。比如，一个Word文档或是一个Windows文件夹对于大部分人来说是无害的，安全的，我们利用了它们的一些可访问网络资源的特性，改造成蜜签，当有黑客从被攻击的服务器或蜜罐中下载后打开文档或者是进入文件夹的时候就会触发告警。

诱饵

- 诱惑攻击者 用于迷惑攻击者的数据，包括文件、数据库、配置、日志、代码等信息，诱使攻击者对蜜罐实施攻击。

Agent

- 攻击流量透明转发 agent通过绑定转发仿真服务蜜罐，将TCP、UDP、ICMP、SYN等类型的攻击流量透明、无感知的被转发到蜜网中的蜜罐里，此时的蜜罐可以捕获攻击者多种操作行为，同时发现新的攻击方式和漏洞利用方法
- 诱饵、密签下发 agent接收下发策略，把策略管控端的诱饵、密签下发到服务器，诱捕引导黑客攻击到蜜网的蜜罐

协议转发

- 协议代理转发 agent通过绑定转发仿真服务蜜罐，攻击者的攻击流量将会通过对应的网络协议（比如HTTP、SSH等）被转发到蜜网中的蜜罐里，此时的转发代理可以实时获取攻击命令、目录爆破探测、登录爆破、ssh异常连接等攻击行为

透明转发

- 探针服务器到蜜网服务器的端口转发

最后更新: 2021年6月27日

2.11 环境准备

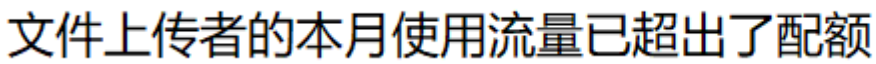
- 一台Linux服务器
- 系统要求:CentOS 7
- 配置要求:2核 4G内存 可用磁盘空间10G以上

2.12 部署安装

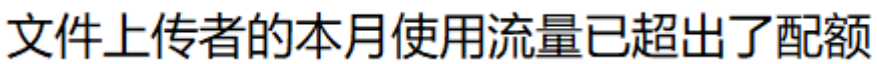
```
1. # 克隆代码实例
git clone https://github.com/seccome/Ehoney.git
2. # cd 到程序目录，执行一键启动脚本
cd Ehoney && chmod +x quick-start.sh && ./quick-start.sh
3. # 【输入序号选择服务器内网IP】，等待约20分钟(视网络情况而定)后安装成功。

安装过程如有任何问题，请看FAQ或ask@seccome.com
```

2.13 安装成功

如看到"Please visit url: xxxx"文案提示，则证明安装成功。




```
打开浏览器后,输入该URL
http://服务器IP:8080/decept-defense
默认用户名:admin，密码:123456。
```



最后更新: 2021年6月27日

2.14 Ehoney快速使用指南

2.14.1 使用说明

1. 登录系统。

   a. 默认登录地址即成功安装后提示地址，如http://192.168.22.176:8080/decept-defense

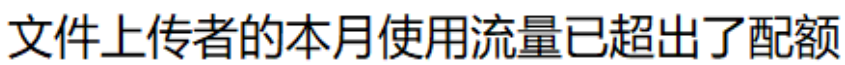
   b. 默认账号admin，密码123456。

2. 新建蜜罐 1. 进入"蜜罐管理"->"蜜罐列表"页面，点击新建。

   a. 输入蜜罐名称(小写字母+数字)、选择镜像类型，点击添加。如:蜜罐名称tomcat，镜像类型为103.39.213.38/ehoney/tomcat:v1。

   b. 等待蜜罐新增成功。(第一次要从harbor拉去镜像，可能耗时3分钟)。

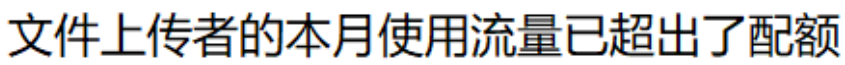
   验证：列表中状态显示“在线”；点击网络探测显示蜜罐正常运行中。

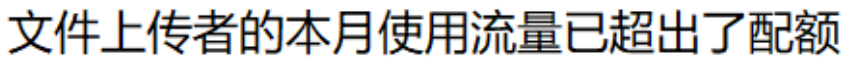


2. 部署密签和诱饵(非必须)

   a. 进入"蜜罐管理"->"蜜罐列表"->"密签列表"，新建选择密签类型(当前仅支持file类型)和密签文件以及部署位置。

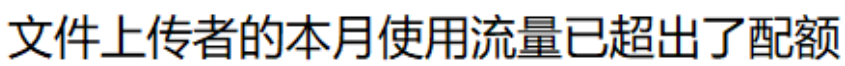
   验证： 列表中状态显示“创建成功”;点击下载后，用office软件打开后在列表上点击详情，展示打开文件的跟踪记录。



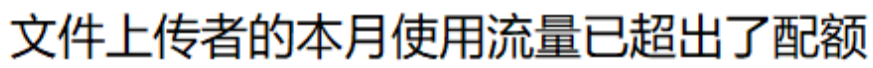


   b. 进入"蜜罐管理"->"蜜罐列表"->"诱饵列表"，新建选择诱饵类型和诱饵文件以及部署位置。

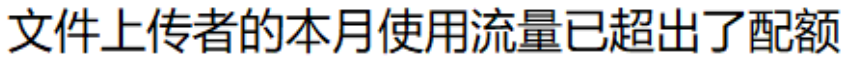
   验证： 列表中状态显示“创建成功”;



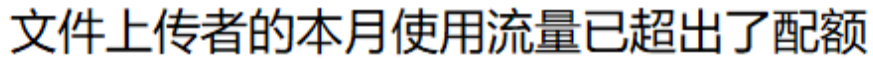
3. 部署探针服务器(非必须)
   a. 点击右上角的下载支持，将压缩包拷贝到即将部署的服务器上。
   b. 解压文件tar -zxvf decept-agent.tar.gz
   c. 修改解压目录中的conf目录下的agent.json，修改strategyAddr参数ip为redis地址的ip，默认安装为当前服务器ip。修改sshKeyUploadUrl的ip尾web服务的ip，默认安装为当前服务器ip。
   d. 执行 chmod +x decept-agent && ./decept-agent -mode=EDGE
   e. 查看启动日志和探针列表是否有此探针服务确认启动是否正常。



4. 建立协议转发
   a. 进入"影子代理"->"协议转发"，选择服务类型(即第2步中新建蜜罐的类型)，选择蜜罐及转发端口(端口范围:0-65535)。
   验证：列表中状态显示“创建成功”;点击网络探测显示正常。



5. 建立透明转发
   a. 新建转发选择蜜罐、协议转发端口、要新建转发的探针主机、转发的蜜罐和端口以及透明转发端口(端口范围:0-65535)。
   验证：列表中状态显示“创建成功”;点击网络探测显示正常; 模拟攻击产生攻击日志，具体请参考这里



2.14.2 进阶使用指南

镜像源配置

1. 搭建好本地或公网上harbor，并上传镜像到harbor项目里。具体可以参考:搭建企业级私有仓库harbor-V2.0并上传镜像
2. 进入镜像源配置，填写harbor地址、用户名、密码、项目名称。

验证：进入镜像列表，列表一刷新，显示自定义蜜罐镜像。

镜像列表

1. 默认seccome harbor源镜像无法修改。
2. 点击编辑可以修改自定义蜜罐端口(诱导黑客攻击的端口)、蜜罐类型(协议代理类型)、操作系统类型。

密签配置

注意：密签跟踪地址理论上要设置成公网IP地址，端口5000。测试环境可以使用默认地址不需要修改。

最后更新: 2021年6月27日

2.15 欺骗防御架构图

欺骗防御由探针、蜜网、管理端三大部分组成：

- 探针Agent，支持透明代理管理，诱饵、密签安装(业务服务器上安装)
- 管理端，支持密网管理、探针管理、诱饵管理、密签管理，攻击溯源
- 密网，支持通过容器API向(蜜罐)部署诱饵和密签 ；支持动态创建容器，销毁容器等。支持协议代理管理

部署欺骗生产部署需要做网络访问控制：

- 探针服务器可以向蜜网发送网络请求，为防止黑客在攻击蜜罐后横向渗透探针，蜜网无法向探针发送网络请求。
- 蜜网服务器可以向蜜罐发送请求，蜜罐不允许访问蜜网服务器。
- 管理平台可以通过容器的API访问蜜罐，蜜罐不允许访问管理平台。

2.16 欺骗防御网络拓扑图

欺骗网络采用分布式部署

- 探针Agent，部署在业务服务器的Agent，通过探针管理透明代理、密签、诱饵，支持动态策略，通过redis更新策略
- 透明代理，部署在业务主机上(探针的一个模块)，模拟业务端口
- 协议代理，部署在蜜罐与透明代理之间，记录攻击行为
- 欺骗网络，基于云原生k3s构建欺骗网络，可以是单台服务器也可以是服务器集群，可以根据实际场景进行部署。在蜜网内的服务器上部署探针Agent，可以通过协议代理技术，将透明代理转发过来的流量，根据协议类型代理到蜜罐中去，使黑客最终攻击的是蜜罐环境。蜜网内的服务器还部署了falco监控模块，可以监控蜜罐中文件的变化，及时取出黑客攻击产生的文件进行监测，分析是否有病毒或者新的木马样本。

2.17 欺骗防御技术全景图

最后更新: 2021年6月29日

2.18 常见问题

2.18.1 FAQ

一 安装问题

1. 服务器环境问题

```
1. 仅支持centos7
2. 磁盘空间必须大于10GB
```

1. 安装问题

```
1. 如果出去端口占用优先确定以及关闭清除之前部署的相关docker容器、然后使用lsof命令查看是否由其他进程占用
2. 镜像拉取失败原因可能由于网络和网速的问题、请确保网络配置正确
3. 启动quick-start.sh 如果出现编码问题，请安装dos2unix进行编码后再执行
```

1. 探针安装

```
1. 如果探针启动后后端探针列表查看不到，需要确定探针目录下的conf/agent.json的Redis配置是否正确
```

二 使用问题

1. 创建蜜罐问题

```
1. 首次安装会较慢，原因是需要从harbor服务器上拉取容器镜像。
2. 镜像拉取失败，可能原因为修改了harbor地址，但是没有在docker中修改，需要修改 /usr/lib/systemd/system/docker.service，配置ExecStart=/usr/bin/dockerd --insecure-registry=（harbor的ip）
```

1. 透明代理问题

```
1. 透明代理下发后创建失败 请查看探针目录下的proxy/log/proxy.log 查看日志确定是否因为端口被占用导致的。
```

1. 伪装代理问题

```
1. 伪装代理无法下发成功
   1.1 登录蜜网服务器，进入/home/ehoney_proxy/目录，查看代理模块是否具有执行权限。
   1.2 请查看项目部署机器的/home/relay/agent/proxy/log/proxy.log 确实是否因为端口占用或端口过大导致。
   1.3 请查看项目部署机器的/home/relay/agent/conf/agent.json的Redis配置是否正确
```

1. 探针问题

```
1. 如果探针启动后探针列表未出现探针数据， 请确定探针目录的下conf/agent.json 中的redis 配置时候正确。
2. 如果探针启动出现权限问题，请使用root权限启动。
```

1. 密签问题

```
1. 密签无法下发到蜜罐，原因可能是该蜜罐镜像不支持/bin/sh或者无法执行wget命令
2. 密签文件打开后无法跟踪，原因可能是密签追踪url配置有问题，可以访问系统设置-密签配置，进行确认。
3. 确认密签追踪url配置方法：访问该url，加上/api/health，如果返回是200，说明部署成功。
```

1. 诱饵问题

```
1. 诱饵无法下发到蜜罐，原因可能是该蜜罐镜像不支持/bin/sh或者无法执行wget命令
```

1. 攻击溯源问题

```
1.暂无
```

三 其他问题

1. 一键安装成功，但是浏览器无法访问？
   - 检查docker容器状态是否正常
   - 检查web服务器容器日志 `docker logs -f $(docker ps  | grep decept-defense:latest | awk '{print $1}')`
1. 为什么模拟攻击没有攻击日志？
   - 通过透明代理列表，网络探测检查攻击IP、端口可以访问。
   - 确保密网服务器上/home/ehoney_proxy目录下对应的代理文件存在。
1. 为什么我部署了多台探针Agent，但是探针列表只显示一个？
   - 删除agent/conf目录下agent文件，重新启动agent进程，确保每台服务器上agentID不一致。
1. 为什么蜜罐列表中新建蜜罐失败？
   - 检查密网服务器资源状态，有可能是磁盘、cpu利用率满了。一般一台2核4G服务器上最多支持部署20个蜜罐。
   - 如果访问harbor不是https协议，需要在docker启动的时候指定insecure-registry。可以通过 `kubectl describe pod` 查看具体报错信息。
1. 为什么蜜罐拓扑图不显示蜜罐和相关连线？
   - 蜜罐拓扑图中是以协议代理、透明代理为维度，如果没有代理则不会显示相关蜜罐。
1. 为什么我部署了探针但是探针列表却不显示？
   - 检查agent中配置是否正确。主要检查conf/agent.json中strategyAddr(redis地址)、strategyPass(redis密码)、sshKeyUploadUrl(服务端更新sshkey地址)。
   - 确认探针服务器和web/蜜网服务器网络可以通信。
1. 为什么协议代理创建失败？
   - 查看协议代理日志/home/relay/agent/proxy/log/proxy.log
   - 检查转发端口是否被占用
   - proxy协议转发二进制文件是否被篡改

最后更新: 2021年6月29日

2.19 贡献者(排名不分先后)

@xuanxiao @yinxue @sihan @gaofeng @yaoyi @afei

最后更新: 2021年6月27日

3. 入门

3.1 生产安装

3.1.1 环境准备

- centos7 至少 4核 4G内存 10Gb磁盘

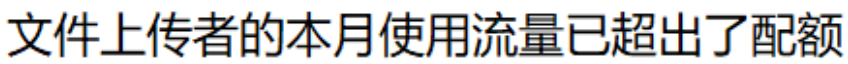
3.1.2 项目下载

- git clone https://github.com/seccome/Ehoney.git

3.1.3 项目启动

启动

- cd Ehoney && chmod +x quick-start.sh && ./quick-start.sh
- 根据提示选择本机使用的IP地址
- (部署过程根据网速以及机器条件大概在5-10分钟之间)



3.1.4 透明代理部署

- 启动完成后点击页面右上方 **下载支持** 按钮下载透明代理Agent
- 解压Agent压缩包、并修改conf/agent.json配置文件为合适的redis地址以及后端地址
- ./decept-agent & 完成启动
- 在**诱捕管理**下的**探针列表**查看

最后更新: 2021年6月27日

3.2 模拟攻击

3.2.1 确定IP及端口

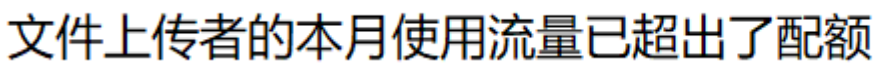
- 进入Ehoney系统，影子代理-透明代理列表
- **攻击IP**：即列表中的**探针IP**
- **攻击端口**： 即列表中的**转发端口**

注意：当未部署探针服务器时，攻击IP为协议转发列表中密网IP，攻击端口为协议转发列表中转发端口

3.2.2 模拟攻击

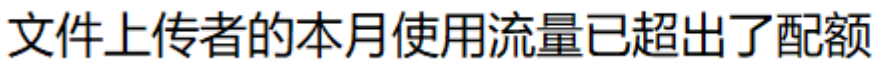
HTTP

`curl http://攻击IP:攻击端口`



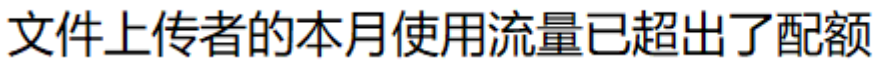
SSH

`ssh root@攻击IP -p 攻击端口` **默认账号密码 root root**



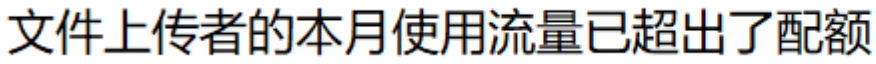
MySQL

`mysql -u root -h 攻击IP -P 攻击端口 -p 123456 -e "show databases"` **默认账号密码 root root**



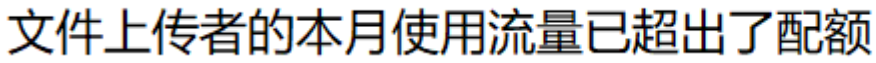
Redis

`redis-cli -h 攻击IP -p 攻击端口` **默认密码 123**



Telnet

`telnet 攻击IP 攻击端口` **默认账号密码 root admin**



3.2.3 其他

- 注意:**模拟攻击类型要和蜜罐类型对应**
- 使用可视化界面进行连接，也会发起攻击日志。如HTTP可以使用浏览器访问;SSH可以使用Putty工具；MySQL可以使用Navicat工具;Redis可以使用RedisDesktopManager工具。
- 可以进入Ehoney系统，告警管理-告警列表好看是否攻击成功。

最后更新: 2021年6月27日

3.3 如何贡献代码和文档

开始之前

github或社区提交问题 欢迎为项目贡献任何代码或文档，但是建议先在github或社区上提交一个问题，和大家共同讨论。

签署贡献者许可协议(CLA)

什么是CLA？

签署协议链接：seccome inc. Contributor License Agreement

单击按钮**Sign in with GitHub to agree**签署协议。

如果有任何问题，请发送邮件至ask@seccome.com。

修改单篇文档

Ehoney文档以Markdown语言编写。单击文档标题右侧的铅笔图标即可提交修改建议。该方法仅适用于修改单篇文档。

批量修改或新增文件

该方法适用于贡献代码、批量修改多篇文档或者新增文档。

Step 1：通过GitHub fork仓库

ehoney项目有很多仓库，以gh-pages仓库为例：

访问https://github.com/seccome/Ehoney/tree/gh-pages。

在右上角单击按钮Fork，然后单击用户名，即可fork出以gh-pages仓库。

Step 2：将分支克隆到本地

定义本地工作目录。

1.定义工作目录。

```
working_dir=$HOME/Workspace
```

2.将user设置为GitHub的用户名。

```
user={GitHub用户名}
```

3.克隆代码。

```
mkdir -p $working_dir
cd $working_dir
git clone https://github.com/$user/ehoney-graph.git
# 或：git clone git@github.com:$user/ehoney-graph.git

cd $working_dir/ehoney
git remote add upstream https://github.com/seccome/ehoney-graph.git
# 或：git remote add upstream git@github.com:seccome-inc/ehoney-graph.git

# 由于没有写访问权限，请勿推送至上游主分支。
git remote set-url --push upstream no_push

# 确认远程分支有效。
# 正确的格式为：
# origin    git@github.com:$(user)/ehoney-graph.git (fetch)
# origin    git@github.com:$(user)/ehoney-graph.git (push)
# upstream  https://github.com/vesoft-inc/ehoney-graph (fetch)
# upstream  no_push (push)
git remote -v
```

4.(可选)定义pre-commit hook。

请将ehoney Graph的pre-commit hook连接到.git目录。 hook将检查commit，包括格式、构建、文档生成等。

```
cd $working_dir/ehoney-graph/.git/hooks
ln -s $working_dir/ehoney-graph/.linters/cpp/hooks/pre-commit.sh
```

pre-commit hook有时候可能无法正常执行，用户必须手动执行。

```
cd $working_dir/ehoney-graph/.git/hooks
chmod +x pre-commit
```

Step 3：分支

1.更新本地主分支。

```
cd $working_dir/ehoney
git fetch upstream
git checkout master
git rebase upstream/master
```

2.从主分支创建并切换分支：

```
git checkout -b myfeature
```

> 由于一个PR通常包含多个commit，在合并至master时容易被挤压（squash），因此强烈建议创建一个独立的分支进行更改。合并后，这个分支可以被丢弃，因此可以使用上述rebase命令将本地master与upstream同步。此外，如果直接将commit提交至 master，用户可以需要在master分支使用hard reset，例如： git fetch upstream git checkout master git reset --hard upstream/master git push --force origin master

Step 4：开发

1.代码风格 ehoney Graph采用cpplint来确保代码符合Google的代码风格指南。检查器将在提交代码之前执行。

2.单元测试要求 请为新功能或Bug修复添加单元测试。 构建代码时开启单元测试 详情请参见使用源码安装ehoney Graph。

> 请确保已设置-DENABLE_TESTING = ON启用构建单元测试。

3.运行所有单元测试 在ehoney根目录执行如下命令：

```
cd ehoney/build
ctest -j$(nproc)
```

Step 5：保持分支同步

```
# 当处于myfeature分支时。
git fetch upstream
git rebase upstream/master
```

在其他贡献者将PR合并到基础分支之后，用户需要更新head分支。

Step 6：Commit

提交代码更改：

```
git commit -a
```

用户可以使用命令--amend重新编辑之前的代码。

Step 7：Push

需要审核或离线备份代码时，可以将本地仓库创建的分支push到GitHub的远程仓库。

```
git push origin myfeature
```

Step 8：创建pull request

1.访问fork出的仓库https://github.com/\(user/ehoney-graph (替换此处的用户名\)user)。

2.单击myfeature分支旁的按钮Compare & pull request。

Step 9：代码审查

pull request创建后，至少需要两人审查。审查人员将进行彻底的代码审查，以确保变更满足存储库的贡献准则和其他质量标准。

添加测试用例

添加测试用例的方法参见How to add test cases。

捐赠项目

Step 1：确认项目捐赠Step 1：确认项目捐赠

通过邮件、微信、等方式联络ehoney官方人员，确认捐赠项目一事。项目将被捐赠至ehoney组织下。

邮件地址：ask@seccome.com 微信：seccome

Step 2：获取项目接收人信息

由ehoney官方人员给出ehoney的项目接收者ID。

Step 3：捐赠项目

由您将项目转移至本次捐赠的项目接受人，并由项目接收者将该项目转移至ehoney组织下。捐赠后，您将以Maintain角色继续主导社区项目的发展。

GitHub上转移仓库的操作，请参见Transferring a repository owned by your user account。

最后更新: 2021年6月27日

4. 教程

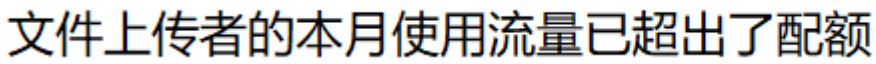
4.1 构建密网

4.1.1 概念

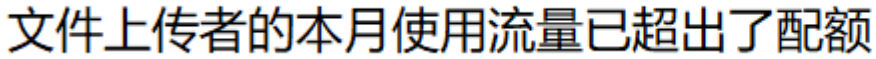
蜜网是由单个或者多个部署了K3S的服务器组成的，蜜网中的服务器支持通过管理平台部署蜜罐，可以仿真企业内网的真实环境。

4.1.2 使用说明

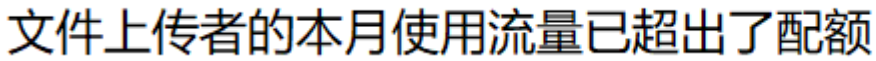
创建蜜罐



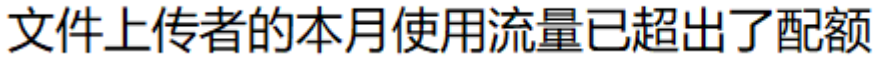
输入蜜罐名称和镜像地址，新建蜜罐



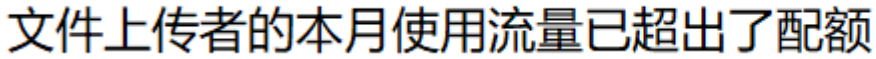
创建协议代理



选择服务类型，选择对应服务的蜜罐，输入需要转发的端口（建议1025-65535范围内的端口号）

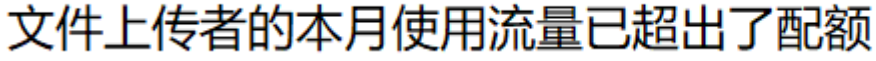


部署探针



通过下载支持，下载agent，在业务服务器上部署探针后，探针会自动注册上线

透明转发



通过蜜罐、协议代理转发端口选择已经创建，且在线的协议代理信息，选择已经上线的探针，输入需要转发的端口创建透明代理。

模拟黑客链路

4.1.3 实现设计

- 通过K3S的云原生技术，部署蜜网服务器。
- 支持只部署Master节点的单机模式，也支持部署多个Worker节点的集群模式。可以根据实际场景进行部署。
- 在蜜网内的服务器上部署agent，可以通过协议代理技术，将透明代理转发过来的流量，根据协议类型代理到蜜罐中去，使黑客最终攻击的是蜜罐环境。

最后更新: 2021年6月27日

4.2 诱饵

4.2.1 概念

- 欺骗防御系统支持多种类型的诱饵下发。
- 包括文件类型的诱饵：比如SQL文件，应用配置文件，备份代码文件，目录缓存文件等文件诱饵。
- 包括history修改的策略诱饵等。

4.2.2 使用说明

- 通过管理平台将诱饵模板下发到探针环境或者蜜罐环境中，可以在黑客攻击的各个环节中进行部署，引诱黑客更深入的攻击蜜罐组成的网中。

4.2.3 实现设计

在攻击者常访问的敏感文件（比如SQL文件，应用配置文件，备份代码文件，目录缓存文件等）放置存在诱惑力的信息，包括但不限于蜜网环境中的ip信息，数据库信息，备份代码信息，系统账号密码信息等。
在容易被攻击的探针上下修改history文件，从而在history中插入攻击者感兴趣的信息，包括但不限于蜜网环境中的ip信息，数据库连接信息，SSH登录账号密码信息等。

最后更新: 2021年6月27日

4.3 密签

4.3.1 概念

- 欺骗防御系统支持多种密签下发。
- 包括文件类型的密签。
- 包括邮箱类型的密签。
- 包括应用类型的密签。
- 包括图片类型的密签。
- 包括DNS类型的密签。
- 包括比特币类型的密签。
- 包括目录类型的密签。
- 包括Git类型的密签。
- 包括SQL类型的密签。

4.3.2 使用说明

- 通过管理平台将密签模板下发到探针环境或者蜜罐环境中，可以在黑客攻击的各个环节中进行部署，当黑客在打开密签的时候，管理平台就能获取黑客的真实攻击ip等信息，对黑客进行特征采集。

4.3.3 实现设计

一种跟踪技术，包括文件、邮箱、git之类的载体，当攻击者打开或者访问的密签的时候，触发对应的策略，服务端主动感知该密签被访问，从而定位攻击者的信息。

最后更新: 2021年6月27日

4.4 伪装代理

4.4.1 概念

- 伪装代理是由部署在业务服务器上的探针Agent实现。
- 能够开启业务相似的端口并能通过协议**代理**把攻击流量转发至对应的蜜罐中，实现诱捕功能。
- 能够部署文件诱饵以及蜜签

4.4.2 使用说明

1. 创建透明转发

- 在**影子代理**下的**透明转发**列表点击**新建**按钮，选择对应的**蜜罐**、**协议代理**、**探针**,最后输入需要转发的端口，并点击确认按钮创建。

4.4.3 实现设计

- 是探针agent中的一个模块，由透明代理agent管理。
- 通过Redis消息订阅模式接收服务端下发透明代理策略触发透明代理的启动。
- 根据透明代理策略启动对代理端口监听以及对代理IP和端口进行转发功能的程序。

最后更新: 2021年6月27日

4.5 协议代理

4.5.1 概念

- 协议代理go语言开发并编译的可执行文件。
- 通过协议代理的agent部署，能够实现开启业务相似的端口并能把攻击流量直接转发至对应的蜜罐中，在转发的过程中能够记录流量信息、解析以及上传攻击告警日志记录

4.5.2 使用说明

1. 部署

- 协议代理agent的与透明代理agent是一样的。区别在于启动参数，协议代理的启动命令为 nohup ./decept-agent -mode RELAY &
- 协议代理agent默认有quick-start.sh脚本自动部署/home/relay/ 目录下
- 协议代理的可执行文件默认有quick-start.sh脚本自动部署在 /home/e-honey/ 目录下

2. 自定义

可在**系统设置**目录下的**协议配置**页面 上传自定义的协议代理可执行文件 可执行文件需支持-backend(蜜罐地址)、-bind(端口绑定地址) -seccenter(后端日志上传地址)参数。例:

```
./sshproxy -backend 10.24.10.2:8080 -bind :8088 -seccenter 192.168.2.11:8080
```

4.5.3 实现设计

1、通过监听服务器上的某个端口，获取请求该端口的流量 2、对请求包进行对应协议的解析，将流量内容按对应的协议的请求体格式进行解析 3、解析出代理信息、攻击者IP、当前攻击时间等，将内容通过服务端的日志接口发送到服务端

最后更新: 2021年6月27日

4.6 云原生支持

4.6.1 概念

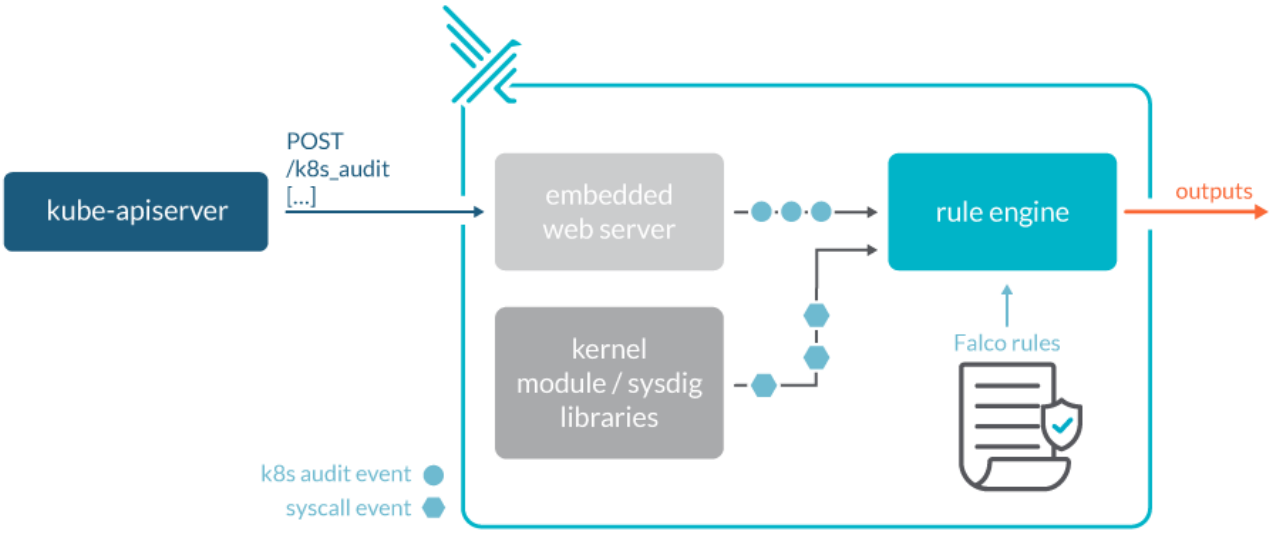
4.6.2 使用说明

Ehoney对云原生的支持主要是通过K3s以及Falco实现的

- 蜜罐作为Pod容器部署在K3s上、而Falco就是为云原生容器安全而生，可以通过灵活的规则引擎来描述任何类型的主机或者容器的行为和活动。
- Ehoney 通过自定义Falco检测规则实现对K3s上的任意容器进行监测，发现可疑的文件操作都将进行上报，并配合服务端对容器内部的文件拷贝操作能够获取到攻击者的攻击脚本样例和操作过程等信息。

Falco 是什么？

- Falco 是云原生的容器运行时（Runtime）安全项目，主要有 Sysdig 为主开发，为 CNCF 项目
- Falco 可以轻松使用内核事件，并使用 Kubernetes 和其他云本机中的信息补充和丰富事件。Falco 具有一组专门为 Kubernetes，Linux 和云原生构建的安全规则。如果系统中违反了规则，Falco 将发送警报，通知到用户



4.6.3 实现设计

- 使用helm方式部署
- 增加自定义规则实现falco能够监听任意目录下的用户创建文件监听
- 修改falco事件日志上报配置，实现falco日志以json格式上报到后端服务

最后更新: 2021年6月27日

4.7 探针Agent

4.7.1 概念

- 用于部署透明转发、蜜签以及诱饵的程序, 有go语言开发并编译的可执行文件。

4.7.2 使用说明

```
1、点击下载支持按钮 下载探针Agent

2、解压agent.zip 并修改agent.json 文件，配置使用的redis信息

3、./decept-agent 启动、并在探针列表页面查看启动的探针信息
```

4.7.3 实现设计

- 协议代理分为管理模块、诱饵模块以及代理模块
- 其中管理模块主要负责心跳、agentid生成以及对诱饵模块和协议代理模块的管理和监控的功能
- 诱饵模块主要负责诱饵下发策略的监听以及部署功能
- 代理模块主要负责代理下发策略的监听以及部署功能

最后更新: 2021年6月27日

4.8 样本捕获与告警

4.8.1 概念

- 样本捕获的是攻击者上传的攻击脚本, 实现的原理为根据K3S上部署的falco容器上报的异常日志并对攻击脚本文件进行docker cp 操作，来获取到攻击者的攻击脚本的。
- falco 相关请参考云原生支持

4.8.2 使用说明

当攻击者在被攻击的蜜罐只能够上传或新建任何文件时，欺骗防御 系统中的falco容器健康到文件变动，会告警到服务端。服务端通过调用K3S的接口，将变动的文件复制到木马检测目录进行检测，以判断攻击者是否上传了木马或者病毒。

4.8.3 实现设计

1、falco容器会监控到蜜罐中的文件变动，一旦发现有蜜罐容器中存在变动的文件，falco容器会向服务端发起告警
2、服务端通过调用K3S的接口，将变动的文件复制到木马检测目录，进行检测

最后更新: 2021年6月28日

5. 定制化指南

5.1 协议代理自定义

5.1.1 功能描述

agent通过绑定转发仿真服务蜜罐，攻击者的攻击流量将会通过对应的网络协议（比如HTTP、SSH等）被转发到蜜网中的蜜罐，此时的转发代理可以实时解析各种类型的协议的数据包获取攻击命令、目录爆破探测、登录爆破、SSH异常连接等攻击行为

5.1.2 开发流程

- 使用Golang、Python等语言开发，流程为：
  1、解析协议请求
  2、获取请求的协议数据包，组成协议包对象，包含属性：
  1）SrcHost（代理信息，比如“:8080”）
  2）SrcPort（代理绑定端口，比如8080）
  3）AttackHost（攻击者IP）
  4）AttackPort（攻击者端口）
  5）DstHost（攻击者IP）
  6）DstPort（攻击者IP）
  7）LocalTime（当前时间）
  8）LogType（协议类型，比如：http）

执行方式为 ./demo -bind :8080 -backend: 192.168.33.38:8080
通过本地8080端口，代理到192.168.33.38的8080端口

5.1.3 使用说明

- 通过web端进行协议转发策略配置，下发到Redis
- 蜜网服务器上的agent接收到转发策略
- agent通过对应的协议代理模块，建立协议转发

5.1.4 注意事项

1、上传的协议代理模块需要有可执行权限 2、需要注意导出的协议代理模块是否适配蜜网服务器操作系统

最后更新: 2021年6月27日

5.2 容器自定义

5.2.1 功能描述

docker镜像是用于k3s创建蜜罐时，需要的容器镜像。

5.2.2 开发说明

- 可以通过自定义Dockerfile，build完之后，导出docker镜像。命令如下：
  1、docker build ： `docker build -t 镜像名 .`
  2、导出docker镜像 ： `docker save -o /xxx/镜像名.tar 镜像名`
  3、将镜像包同步给 ask@seccome.com
  4、欺骗防御开发团队对镜像包进行安全扫描，如果无安全风险，团队会将该镜像包上传到公用Harbor上。

5.2.3 使用说明

- 在系统设置-镜像列表中可以对线上拉取的镜像（除了系统默认提供的镜像）进行编辑，包括端口号，服务类型等
- 在新建蜜罐的时候，拉取配置好的镜像列表，创建蜜罐

5.2.4 注意事项

1、镜像必须支持/bin/sh 2、镜像必须安装wget命令

最后更新: 2021年6月28日

6. 参考手册

6.1 版本 V1.0 - 2021年6月22日

功能	描述	进度
蜜网	基于K3S的容器管理，支持蜜网的生命周期管理	done
探针	实现协议代理、透明转发管理，蜜签、诱饵安装	done
密签	支持file、exe、img、url、git、svn等	doing
诱饵	支持ssh、ftp 、rsync、mysql、wget、nano	done
协议代理	支持高交互ssh、telnet、ftp、http、rdp、mysql、redis、mq、es、hadoop等	done
云原生	支持云原生falco监控，可以记录行为以及攻击样本	done
网络拓扑	能够实现攻击网络拓扑可视化	done

6.2 未来演进

- AI对抗
- 溯源反制
- 联动其他安全产品
- 更丰富的协议(包含iot，工控等)、蜜签、诱饵

最后更新: 2021年6月27日

6.3 蜜罐列表

6.3.1 新建蜜罐

用例编号	测试标题	重要级别	预置条件	测试步骤	预期结果	实际结果
AddHoneyPod_1	正常新建蜜罐	高级别	无	a.新建蜜罐 名称为test b. 选择镜像 c. 点击新建	新建成功	
AddHoneyPod_2	新建重名蜜罐	高级别	已存在名称为test的蜜罐	a. 新建蜜罐 名称为test b. 选择镜像 c. 点击新建	提示同名蜜罐已存在	
AddHoneyPod_3	新建蜜罐无法访问Harbor	高级别	Harbor服务器无法访问	a. 新建蜜罐 名称为test b. 选择镜像 c. 点击新建	新建失败	
AddHoneyPod_4	新建蜜罐镜像不正确	高级别	镜像无常驻进程启动失败	a. 新建蜜罐 名称为test b. 选择镜像 c. 点击新建	新建失败	

最后更新: 2021年6月27日