网络入侵检测课程论文

基于 WebHoney 与 Cowrie 的公网攻击诱捕与行为分析

学号：23121677

姓名：范舒舰

学院：计算机工程与科学学院

专业：网络空间安全

2026 年 6 月

目　录

摘要

本文基于一台真实公网服务器，构建了由 WebHoney、Cowrie、Nginx 分流、Shadow 旁路、统一事件日志和风险评分模块组成的轻量级多入口蜜罐系统。系统在不影响真实 Web 业务的前提下，将公网中的 Web 扫描、敏感路径探测和 SSH 弱口令爆破导入受控环境，并对一个月运行日志进行统计分析。实验结果显示，公网攻击以自动化扫描和 SSH 爆破为主，高风险事件数量相对较少，但能够反映较完整的攻击意图。系统捕获到了敏感文件探测、诱饵凭据利用、假后台登录、路径穿越、SSH 命令执行、脚本下载和公钥持久化尝试等行为。实验说明，在资源有限的服务器环境中，轻量级蜜罐仍能提升攻击行为可观测性，并为风险处置和后续威胁分析提供数据基础。

关键词：WebHoney；Cowrie；蜜罐；公网扫描；SSH 爆破；攻击链分析

1 引言

公网服务器上线后，会持续收到 Web 扫描、敏感文件探测和 SSH 弱口令爆破，这类持续存在的非预期流量可视为公网背景噪声的一部分[5]。普通访问日志能够记录来源 IP、路径和状态码，很难进一步判断攻击者是否存在凭据利用、命令执行、文件下载或持久化尝试。因此，本文将蜜罐作为受控观察环境，用于补充普通日志无法呈现的攻击过程。

2 技术基础与系统定位

2.1 蜜罐与欺骗防御

蜜罐是一种主动防御与欺骗检测技术。它通过构造虚假的系统、服务、页面或凭据，吸引可疑访问者进入受控环境，并记录其交互行为，从而暴露更多攻击意图。欺骗防御强调在真实环境周围布置诱饵资产，使攻击者更容易触发可观测事件。常见诱饵可以包括假配置文件、假后台、假上传入口、假 SSH 终端和假服务端口。

2.2 多入口蜜罐的实验定位

单一蜜罐只能覆盖一个入口，而公网攻击通常同时包含 Web 扫描、SSH 爆破、端口探测和凭据尝试。本文构建一个轻量级多入口诱捕系统：Web 侧通过 Nginx 分流和 WebHoney 诱捕敏感路径；SSH 侧通过 Cowrie 承接对默认 SSH 入口的爆破尝试。

从完整蜜罐体系看，OpenCanary、Dionaea 和 T-Pot 都具有一定参考价值。OpenCanary 可用于模拟 FTP、Telnet、MySQL、Redis、SNMP 等轻量服务，适合发现多协议扫描；Dionaea 更侧重恶意样本、下载 URL 和样本哈希的捕获；T-Pot 则通过 Docker 集成多类蜜罐，并提供统一存储和可视化分析能力。受服务器资源、隔离环境、安全风险和维护成本等因素限制，本文没有实际部署这些系统。

3 系统设计与实际部署

部署服务器同时承载真实 Web 业务和蜜罐诱捕模块。在不影响真实业务的前提下，实际部署范围包括四个部分：第一，基于 Nginx 的统一入口分流，用于区分真实业务、显式蜜罐站点、高价值路径和普通异常访问；第二，WebHoney Web 蜜罐，用于捕获敏感文件探测、诱饵凭据利用、后台登录、上传和文件操作；第三，Cowrie SSH 蜜罐，用于承接对默认 SSH 入口的爆破尝试，并记录登录后的命令行为；第四，统一日志与风险处置模块，用于归并 WebHoney、Cowrie、Nginx catch-all 和 shadow 日志，并根据风险分进行临时封禁或记录。

3.1 总体架构

系统总体架构如图1所示。整体采用“真实业务隔离、蜜罐入口分流、统一日志分析”的结构。本地 Flask 服务上，由 `shopping.vertexf.top` 对外提供真实业务；由 `honey.vertexf.top` 对外暴露蜜罐服务；SSH 管理入口迁移到非默认端口，仅供管理员使用；公网默认 SSH 入口则由 Cowrie 蜜罐承接。

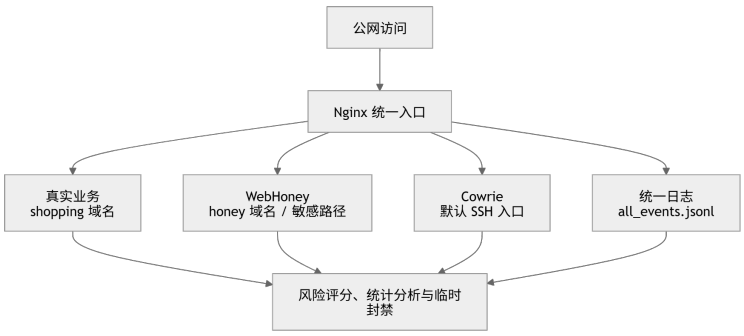

图 1: 总体架构示意

系统根据访问入口和路径特征进行分级处理。普通异常路径可以直接由 Nginx 返回 444；敏感路径和高价值行为则进入 WebHoney ；SSH 爆破则进入 Cowrie 的伪终端环境。

3.2 Nginx 入口分流与隔离策略

Nginx 是系统入口层的核心。负责反向代理真实业务和 WebHoney，同时承担第一层流量筛选功能[2]。按访问域名、访问路径和请求特征进行分流以减少蜜罐负载。

系统中的 Web 入口可以分为四类。第一类是真实业务入口，即 `shopping.vertexf.top`，主要转发到真实商城服务。第二类是显式蜜罐入口，即 `honey.vertexf.top`，访问者进入后可以看到伪造页面、假配置、假后台和上传中心。第三类是直接 IP 下的高价值路径，例如 `/.env`、`/.git/config`、`/admin`、`/phpmyadmin` 等，被导向 WebHoney。第四类是普通异常路径，例如随机字符串、畸形请求或无意义扫描，这类请求主要返回 444 或普通错误页面，只做低风险统计。

Shadow 旁路是本系统中的一个特殊设计。对于真实业务域名下的敏感路径，前台保持原业务响应，同时，后台单独记录来源 IP、访问路径、User-Agent 和时间。这样可以

避免攻击者明显感知到蜜罐存在，也能统计真实业务面受到的探测压力，模拟真实环境中的隐蔽检测点。

3.3 WebHoney 诱捕模块设计

WebHoney 主要包含假配置文件、假后台和文件管理/上传入口。

第一类诱饵是假配置文件。公网扫描中，`.env`、`.git/config`、配置备份文件和云密钥路径经常被自动化工具探测。WebHoney 在这些路径上返回伪造内容，例如后台账号、数据库账号、API Key 和内部路径。这些内容只用于观察访问者是否继续利用。

第二类诱饵是假后台。页面如图2所示。系统提供 `/admin` 入口。若访问者读取配置后继续尝试登录后台，系统会记录 `sandbox_admin_login` 或 `honeycred_used` 事件。

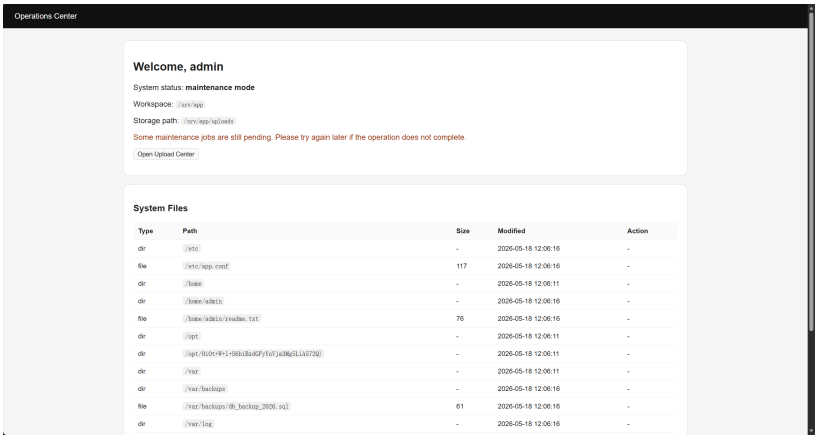

图 2: 假后台页面

第三类诱饵是文件管理与上传功能。登录假后台后，页面会展示伪装目录和文件。下载操作可以被记录，但编辑和删除操作不会真正修改真实文件，上传功能会检查文件名、扩展名和内容特征。对于 PHP 脚本、WebShell 特征、PE 或 ELF 特征文件，系统只记录高危上传事件，不让文件进入真实业务目录。

3.4 Cowrie SSH 蜜罐部署

真实 SSH 日志可以记录登录失败、无效用户和断开连接，但无法观察攻击者登录后会执行什么命令。本系统将真实 SSH 管理入口迁移到非默认端口；公网默认 SSH 入口由 Cowrie 承接，用于记录爆破和登录后的行为。

Cowrie 是一个中交互 SSH 蜜罐[1]。攻击者连接后进入伪造 Linux Shell，系统记录会话建立、客户端信息、用户名密码、登录结果、命令输入、文件下载和会话关闭等事件。由于交互发生在模拟环境中，攻击者操作不会影响真实服务器。

本文在 Cowrie 中配置了伪文件系统，使交互环境更接近真实服务器。攻击者成功登录后看到的是 Cowrie 构造的伪终端环境，如图3所示。伪文件系统中包含 `/srv/app`、`/srv/app/.env`、`/etc/app.conf`等诱饵内容。攻击者如果执行读取配置文件、查看系统信息、下载脚本等操作，

```
C:\Users\23049>ssh root@198.44.178.194
root@198.44.178.194's password:
root@admin_panel:~# cd /etc
root@admin_panel:/etc# ls
app.conf group    passwd
root@admin_panel:/etc#
```

图 3: 成功登录Cowrie页面

都会进入 Cowrie 日志。把普通的 SSH 登录成功事件进一步转化为命令行为分析。

在后续一个月的日志中，Cowrie 捕获到大量弱口令爆破、登录成功和命令输入行为。特别是攻击者尝试写入 `authorized_keys`、使用 `chattr +ai` 设置不可变属性，观察到了入侵后的环境探测和持久化尝试。

3.5 统一日志与风险评分

系统设计了统一事件采集模块。事件采集器持续读取各类日志，将其转换为统一 JSONL 格式，并写入统一事件文件。每条事件通常包含时间、来源模块、来源 IP、请求方法、访问路径、事件类型、风险分、User-Agent 和扩展字段。

风险评分模块基于统一事件进行周期性计算。当某个 IP 在短时间内累计风险达到阈值时，系统会通过 ipset 执行临时封禁，并把处置结果写入封禁日志。

4 日志统计与分析

4.1 日志来源与处理方式

本节分析的数据来自系统运行一个月期间产生的真实日志。

主分析覆盖 1,187,391 行日志，提取出 3,861 条高风险事件；Cowrie 原始日志覆盖 731,595 行，提取出 1,440 条高风险 SSH 行为；Nginx access 日志覆盖 45,221 行，提取出 2,485 条高风险 Web 行为。整体上，日志以 SSH 会话事件、登录失败、Nginx 444 和敏感路径访问为主，说明公网背景噪声主要来自自动化 SSH 爆破和 Web 随机扫描。

4.2 SSH 攻击行为分析

Cowrie 原始日志显示，SSH 主要攻击形态是自动化弱口令爆破[6]。高频来源包括 `87.251.64.149`、`176.65.139.91`、`194.59.30.76`、`195.178.110.4`、`84.247.170.63`。这些 IP 的失败次数集中且重复，符合脚本化爆破特征。SSH 高频爆破来源如图4所示。

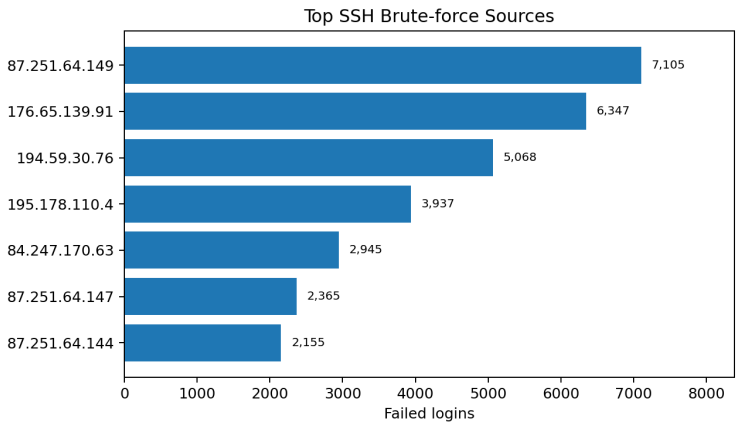

图 4: SSH 高频爆破来源 IP

用户名字典以默认账号和服务账号为主。`root` 出现次数最高，其次包括 `admin`、`user`、`ubuntu`、`support`、`test`、`postgres` 等。`vertexf` 也被多次尝试，说明部分扫描器会根据主机名或域名生成候选账号。密码字典以 `123456`、`password`、`admin`、`root` 等通用弱口令为主，同时出现 `Wangsu@2017`、`welltech12` 等固定特殊密码，表明部分流量

来自固定工具或固定字典。

登录成功后的命令主要分为四类：一是环境探测，如 `ls`、`whoami`、`hostname`、`uname`；二是系统信息读取，如 `cat /etc/passwd` 和 `cat /proc/cpuinfo`；三是诱饵配置读取，如 `cat /srv/app/.env` 和 `cat /etc/app.conf`；四是下载与持久化尝试，如 `wget http://example.com/bot.sh`、写入 `authorized_keys` 和执行 `chattr +ai`。

进一步筛选发现，日志中出现 35 次包含 `chmod +x clean.sh`、`sh clean.sh`、`chmod +x setup.sh` 和 `sh setup.sh` 的组合命令，同时记录到 `wget http://example.com/bot.sh` 和文件下载哈希。部分会话尝试创建 `~/.ssh`、写入 `authorized_keys`，并通过 `chattr +ai` 固化文件，属于典型 SSH 持久化控制尝试[4]。由于未进行样本分析，本文仅将下载行为归为疑似恶意载荷下载。

4.3 Web 攻击行为分析

Nginx access 日志显示，Web 侧请求主要分为正常访问、跳转请求、异常路径请求和敏感路径扫描。状态码中 `444` 数量最高，说明大量直接 IP、未知 Host 或随机路径请求在入口层被静默断开。WebHoney 捕获的关键事件分布如图5所示。

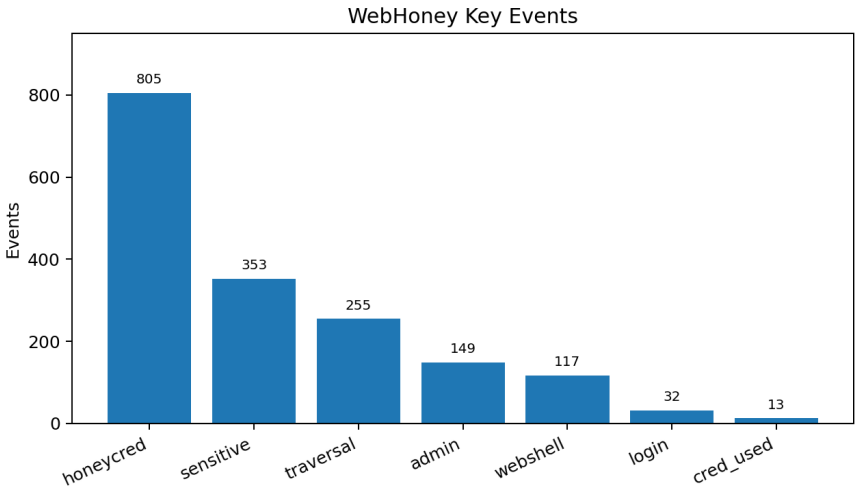

图 5: WebHoney 关键事件分布

Web 扫描路径集中在配置文件、源码泄露、云密钥、后台入口和框架调试页面。典型路径包括 `/.env`、`/.git/config`、`/.aws/credentials`、`/admin`、`/wp-admin/install.php`、`/phpinfo.php` 和 `/_profiler/open`。路径穿越请求中出现 `%2e` 和双重编码，并指向 `/bin/sh`，属于命令执行前置尝试。

系统捕获到大量 `honeycred_exposed`，说明伪配置文件被访问；同时出现 `sandbox_admin_login` 和 `honeycred_used`，说明部分访问者读取诱饵凭据后继续尝试登录假后台。

4.4 风险处置与效果分析

风险引擎基于统一事件日志计算来源 IP 在最近 60 秒和 7 天窗口内的风险分。达到阈值后，系统将该 IP 加入 `lighthoney_ban` ipset 集合，并把执行结果写入封禁日志。

表 1: 典型封禁 IP 记录

IP 地址	封禁次数	总时长/min	最长/min	最高60秒分
87.251.64.149	57	8434	248	22800
87.251.64.147	21	602	42	5510
87.251.64.144	19	525	47	5510
176.65.139.91	2	304	185	174480
84.247.170.63	1	152	152	151145

如表1所示，封禁结果显示两类高风险来源。`87.251.64.149` 属于长期重复攻击源，封禁次数和总时长都较高；`176.65.139.91` 和 `84.247.170.63` 属于短时间高强度来源，单次风险分和封禁时长较高。该结果表明，风险评分可以同时识别持续型爆破和突发型攻击。

系统运行结果表明，统一日志能够从大量背景噪声中提取高价值行为。Web 侧捕获到敏感路径扫描、诱饵凭据使用和假后台操作；SSH 侧捕获到弱口令爆破、命令输入、脚本下载和持久化尝试。当前不足是风险引擎仍采用周期性扫描历史事件的方式，若攻击量再提升一个量级或开放更高交互的蜜罐，会导致日志分析耗用资源过多，后续可改为保存文件 offset 和 IP 状态的增量分析模式。

5 典型攻击链分析

5.1 分布式爆破

结合封禁记录和 Cowrie 日志可以看出，SSH 攻击过程具有分阶段特征。`87.251.64.144/28` 网段内多个相近 IP 反复出现，其中 `87.251.64.149`、`87.251.64.147`、`87.251.64.144` 和 `87.251.64.145` 多次触发临时封禁。该网段共记录到 `cowrie.login.failed` 14525 次，但只有 `87.251.64.149` 出现 1 次登录成功，命中的诱饵账号为 `admin:nimda`。这说明该网段主要承担弱口令爆破和账号验证任务，而不是主要执行入侵命令。

5.2 后续控制尝试

登录成功后的 session 分析进一步说明了后续攻击意图。Cowrie 共记录 1417 个登录成功 session，其中 1357 个登录后没有明显命令，主要表现为账号可用性验证。更高危的是 39 个 session 可视作“建立长期 SSH 后门入口”，主要来源为 `130.12.180.51`。该来

源多次使用 root:nimda 登录后执行相同命令链，包括运行 clean.sh、setup.sh，删除临时脚本，创建 ~/.ssh，写入 authorized_keys，并通过 chattr +ai 固化文件。这类行为具有明确的持久化控制意图。

此外，日志中还出现少量环境探测和载荷下载行为，例如执行 uname、ls、cat /proc/cpuinfo 判断系统环境，以及读取 /srv/app/.env 后通过 wget http://example.com/bot.sh 下载脚本。

6 总结与反思

本文基于真实公网服务器部署了 WebHoney 与 Cowrie 组成的轻量级多入口蜜罐系统，并结合 Nginx 分流、Shadow 旁路、统一事件日志和风险评分模块，对一个月运行日志进行了分析。结果显示，公网攻击以 Web 自动化扫描和 SSH 弱口令爆破为主，高风险事件比例不高，但能够反映较完整的攻击意图。系统捕获到了敏感路径探测、诱饵凭据利用、后台登录、路径穿越、SSH 登录成功、命令执行、脚本下载和 SSH 公钥持久化尝试等行为，说明轻量级蜜罐在资源有限的服务器环境中仍能提高攻击过程的可观测性。后续可以在隔离环境中扩展多协议蜜罐、恶意样本捕获和统一可视化分析能力，使系统从轻量级攻击记录平台进一步发展为更完整的威胁感知与研判平台[3]。

参考文献

[1] Cowrie Project. Cowrie SSH/Telnet Honeypot Documentation.

[2] Nginx Documentation. Logging and Reverse Proxy Configuration.

[3] Deutsche Telekom Security. T-Pot Honeypot Platform Documentation.

[4] MITRE ATT&CK. Enterprise Techniques: Valid Accounts, Command and Scripting Interpreter, Account Manipulation.

[5] Pang R, Yegneswaran V, Barford P, Paxson V, Peterson L. Characteristics of Internet Background Radiation[C]//Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. 2004: 27–40.

[6] Antonakakis M, April T, Bailey M, Bernhard M, Bursztein E, Cochran J, et al. Understanding the Mirai Botnet[C]//Proceedings of the 26th USENIX Security Symposium. 2017: 1093–1110.