ArkHoney:基于协同机制的Web蜜罐

贾召鹏[1),2)] 方滨兴[1),4),5)] 崔 翔[2),5)] 刘奇旭[2),3)]

[1)](北京邮电大学网络空间安全学院 北京 100876)

[2)](中国科学院信息工程研究所 北京 100093)

[3)](中国科学院大学网络空间安全学院 北京 100049)

[4)](电子科技大学广东电子信息工程研究院 广东 东莞 523808)

[5)](广州大学网络空间先进技术研究院 广州 510006)

摘 要 基于Web技术的互联网应用的迅速发展引起了黑客的关注,针对Web的攻击成为互联网上的主要威胁之一.Web蜜罐技术可以帮助人们收集攻击信息从而使得人们能够更好的应对此类威胁,因而受到安全研究人员的重视.然而,蜜罐只能捕获针对自身的攻击,如果攻击者发现想要攻击的应用不在蜜罐系统中,那么攻击者将不会进行下一步动作,蜜罐系统也就不能捕获到攻击数据.为了提高攻击者攻击Web蜜罐成功的概率,文中提出了一种在Web蜜罐系统中部署多个不同应用的方案.首先,提出了蜜罐簇的概念,由多个不同的应用蜜罐组成蜜罐簇;然后设计了蜜罐簇协同算法,通过协同算法使得整个蜜罐簇作为一个Web蜜罐发挥作用;最后使用四种不同的应用实现了基于协同机制的蜜罐原型ArkHoney.在两个月的部署中,ArkHoney蜜罐系统捕获到来自985个不同IP的7933次请求.通过分析捕获到的数据,人工已确认针对四种应用的26次攻击.文中对捕获到的总体数据进行了统计,然后选取蜜罐簇中不同蜜罐捕获到的案例进行分析,实验表明文中提出的基于协同机制的Web蜜罐能有效增加蜜罐系统对攻击的捕获能力.

关键词 蜜罐;蜜罐簇;Web蜜罐;Web应用;协同

中图法分类号 TP393 **DOI号** 10.11897/SP.J.1016.2018.00413

ArkHoney: A Web Honeypot Based on Collaborative Mechanisms

JIA Zhao-Peng[1),2)] FANG Bin-Xing[1),4),5)] CUI Xiang[2),5)] LIU Qi-Xu[2),3)]

[1)](*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing* 100876)

[2)](*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

[3)](*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049)

[4)](*Institute of Electronic and Information Engineering of UESTC in Guangdong, Dongguan, Guangdong* 523808)

[5)](*Cyberspace Institute of Advanced Technology, Guangdong University, Guangzhou* 510006)

Abstract With the rapid development and increasing growth of network services on Websites, Web attack has drawn significant attention from attackers, making it one of the major threats on the Internet. Such attack has caused great loss of financial and intellectual property. High interaction honeypots can attract attackers, detect attacks and suspicious behaviors on the Internet and collect information about what attackers do during and after their attacks. The information collected by a honeypot can effectively help security vendors and services providers to learn the threats websites faced and thus protect websites from attacks. However, what attack information can be collected depend on the type and version of web applications installed in the web honeypot. High interaction Web honeypots can only collect limited information from attacks if the target

收稿日期:2016-06-28;在线出版日期:2017-06-11.本课题得到东莞市引进创新科研团队计划(201636000100038)、国家重点研发计划(2016YFB0801604)资助.**贾召鹏**,男,1988年生,博士研究生,主要研究方向为信息安全、网络安全.E-mail:jzp_jie@163.com.**方滨兴**,男,1960年生,博士,教授,博士生导师,中国工程院院士,主要研究领域为网络安全、信息内容安全.**崔 翔**(通信作者),男,1978年生,博士,研究员,主要研究领域为网络安全.E-mail:cuixiang@iie.ac.cn.**刘奇旭**,男,1984年生,博士,副研究员,主要研究方向为Web安全.

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

application is not deployed in a honeypot, due to the fact that the attacks will failed. In order to increase probability that a Web honeypot will be successfully attacked, It's better to deploy various Web applications in one single Web honeypot. This paper proposes a design scheme for high interaction Web honeypot, intending for the obvious promotion of success probability of a Web honeypot be attacked, so that to enhance attack information collection on high-interaction Web honeypot. First, we analysis the process of Web attacks against honeypot and introduced a concept called honeypot-cluster which consists of several Web honeypots and a cooperative control unit. In each of the Web honeypot, different kinds of Web applications have been installed. Then, a collaborative algorithm is designed. The cooperative control unit uses collaborative algorithm to determine which application in the honeypot-cluster is the attacker' desire. By using the collaborative algorithm, a honeypot-cluster performance as if it is a single Web honeypot. When the honeypot-cluster get an attack, it will forward the attack to the application selected by collaborative algorithm. In this way, a Web honeypot can collect more attack information, because more attacks will succeed. Next, we design and implement a prototype system called ArkHoney, which is based on the collaborative algorithm. ArkHoney is implemented with four different Web applications (Joomla, WordPress, phpMyAdmin, Drupal). We describe its design, implementation, risk control, as well as how to log the attacks in details. We deploy the system for two months to evaluate its performance. In the experiments, ArkHoney attracted 7933 hits, which came from 985 distinct IP addresses. After filter out benign crawler-generated traffic, we were able to identify with 638 different IPs, originating from 47 countries and regions. By analyzing the data collected by ArkHoney, 26 attacks against four applications were manually confirmed. We provide some insights into a few interesting attacks that we identified during the operation of our honeypot. In addition, we also introduce some failed attacks. These attacks failed for the reason that the attackers' desired applications are not in our ArkHoney. Experimental results show our work can enhance attack collection on high-interaction Web honeypot. In the end of the paper, we conclude the shortcomings of ArkHoney and provide ideas on future evaluation of our scheme.

Keywords honeypot; honeypot-cluster; Web honeypot; Web application; collaborative

1 引 言

随着互联网技术的发展,基于 Web 技术的互联网应用越来越广泛.个人通过 Web 平台搭建自己的博客站点,公司在 Web 平台上架设信息系统,社交网络依赖 Web 平台提供服务.根据 NETCRAFT 网站 2015 年 11 月发布的报告①,全球 Web 站点数量已经超过 9 亿个,Web 应用的迅速发展,也引起了黑客的关注,Web 安全形势严峻②.

通过基于 Web 的攻击,攻击者可以获取大量隐私数据③,这造成了巨大的财产损失.在更多的情况下攻击者会和被控制的网站保持访问以将其作为恶意基础设施的一部分,用于诱骗受害者下载恶意软件、发送垃圾邮件、组成僵尸网络等目的,这给网站所有者造成了声誉及法律方面的影响.近期的安全事件表明 Web 平台也开始成为勒索软件的目标④.

严峻的 Web 威胁形势使得很多的研究关注 Web 安全,如恶意 Web 页面检测、Web 应用漏洞挖掘、被入侵 Web 主机检测、高危网站预测等.然而频发的 Web 应用漏洞证明问题远远没有解决,经常能看到如下的场景:一个漏洞被公布,然后漏洞被用于

① Netcraft. Web server survey. http://news.netcraft.com/archives/2015/11/16/november-2015-web-server-survey.html 2015,11,16

② 2015 Web Application Attack Report (WAAR). http://www.impe-rva.com/docs/HII_Web_Application_Attack_Report_Ed6.pdf

③ U.S. says ring stole 160 million credit card numbers. http://dealbook.nytimes.com/2013/07/25/arrests-planned-in-hacking-of-fi%nancial-companies/2013,7,25

④ Cybercriminals have expanded their ransomware attacks to entities that are more likely to pay the money and have started to target companies by encrypting important databases used by their Websites. http://news.softpedia.com/news/Web-Databases-Encrypted-in-New-Ransomware-Scheme-471780 2016

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

攻击，之后解决方案公布. 2014 年 10 月 Drupal 爆出严重 SQL 注入漏洞，漏洞公布 7 小时就出现自动化攻击①；2015 年 12 月 JoomlaCVE-2015-8562 漏洞曝光，该漏洞在漏洞补丁可用之前至少已经被利用了两天②，而根据赛门铁克公司的数据每天有 16 600 次针对该漏洞的扫描攻击③；2016 年 structs2 连续爆出高危漏洞，境外某黑客组织经常利用此类漏洞攻击我国政府网站.

针对这一情况，需要一种积极的应对方式收集攻击者的攻击信息以更好的应对威胁：对于未知漏洞可以发现利用该漏洞的攻击活动，对已知漏洞可以捕获利用该漏洞的信息来生成入侵检测与恶意样本签名以减小危害. 高交互 Web 蜜罐技术正是提供了这样一种手段而受到 Web 安全研究人员的关注[1-2].

然而蜜罐技术的一个缺陷会限制蜜罐对攻击数据的搜集，那就是蜜罐仅能捕获针对自身的攻击，如果攻击者所利用的漏洞没有在蜜罐系统中，那么蜜罐将不能捕获到这次攻击的信息. 例如使用 Joomla 应用搭建的蜜罐系统仅能捕获针对 Joomla 应用漏洞的攻击，如果攻击者想要利用的是 WordPress 应用的漏洞，那么攻击就会失败，蜜罐系统也就会失去这次攻击的详细信息. Web 应用种类繁多，如果采用单一的 Web 应用来部署蜜罐，势必会缩小蜜罐对攻击数据的搜集范围. 为了增强蜜罐系统对攻击的搜集能力，一种理想的方案是在 Web 蜜罐系统中部署多种应用.

本文提出了蜜罐簇的概念，并使用这一概念设计了基于协同机制的蜜罐系统. 在所提方案中多个不同的 Web 应用形成一个蜜罐簇，通过本文提出的协同算法，对于一次攻击在蜜罐簇中选择一个“最合适”的应用蜜罐进行响应，以此增强蜜罐系统对攻击信息的搜集能力. 本文中实现了原型系统，通过实际部署证明了所提方案的有效性.

本文第 2 节介绍相关工作；第 3 节提出蜜罐簇的概念与协同算法；第 4 节介绍原型系统的设计与实现；第 5 节介绍实验结果；第 6 节进行总结并介绍下一步研究计划.

2　相关工作

2.1　蜜罐技术

蜜罐是一类安全资源，其价值在于未授权的利用[3]. 自从 20 世纪八九十年代诞生于安全人员的实践中后[4]，蜜罐技术已被广泛用于入侵检测、僵尸网络追踪、DDoS 防御、恶意代码收集等领域. 蜜罐技术也常常被用来研究新出现的安全威胁，如针对电话骚扰问题，Gupta 等人[5]实现了电话蜜罐 Phoneypot，通过使用 39 696 个电话号码，在 7 周时间内捕获了 130 万次呼叫，通过这些呼叫分析了电话 DoS 等攻击类型. 而针对工控系统的安全问题，趋势科技部署了工控蜜罐④，通过模拟 ICS/SCADA 设备，伪装成一座水压力站，仅仅部署 18 个小时就捕获到第一例攻击，在 28 天的时间里捕获到了来自 14 个国家的 39 次攻击.

根据蜜罐与攻击者之间的交互能力，蜜罐可以分为高交互蜜罐与低交互蜜罐两类. 低交互蜜罐一般采用仿真的方式模拟系统或者服务有漏洞的部分，如 Honeyd[6]可以模拟操作系统和服务，郭军权等人提出的 Spampot[7]实现了对邮件系统的模拟，用于捕获垃圾邮件. 低交互蜜罐只能捕获攻击流量而不能被真正控制，这一类蜜罐能力有限，容易被识别，同时不能捕获完整的攻击过程，对未知攻击应对能力弱；但是因为不能被真正攻陷所以安全性高，方便部署和管理. 而高交互蜜罐使用真实的系统和服务来搭建蜜罐，提供给攻击者一个可以攻陷的真实的环境. 比较典型的高交互蜜罐是国际蜜网组织提出的 Honeynet[8]，Honeynet 在一个可控的环境中部署存在漏洞的真实系统，而 HoneyBow[9]则利用真实的系统实现了一个自动的恶意代码捕获器，通过文件系统实时监控和文件列表交叉对比方法捕获恶意代码样本. 高交互蜜罐可以捕获针对未知漏洞的攻击，同时能收集到完整攻击过程的信息，但是常常需要较高的设置和维护代价，而且被攻陷后也会带来较高的风险. 为了节约成本高交互蜜罐常常会使用虚拟机部署.

2.2　Web 蜜罐

随着 Web 威胁的兴起，蜜罐技术也被用于研究

① Drupal Core-Highly Critical-Public Service announcement-PSA-2014-003. https://www.drupal.org/PSA-2014-003 2014,10,29

② Critical 0-day Remote Command Execution Vulnerability in Joomla. https://blog.sucuri.net/2015/12/remote-command-execution-vulnerability-in-joomla.html 2015,12,14

③ Vulnerable Joomla! Installation under active attack. http://www.symantec.com/connect/blogs/vulnerable-joomla-installation-under-active-attack 2015,12,24

④ Who's Really Attacking Your ICS Equipment?. http://www.trendmicro.com.hk/cloud-content/apac/pdfs/security-intelligence/white-papers/wp-whos-really-attacking-your-ics-equipment.pdf 2013

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

Web 安全. Web 蜜罐分为两类:客户端蜜罐和服务端蜜罐. 客户端蜜罐通过主动访问网站来检测恶意活动,最早出现的这类蜜罐工具之一是 honeymonkey[10]. 服务端蜜罐通过暴露有漏洞的服务来吸引攻击着. 本文主要关注服务端蜜罐.

与传统的蜜罐一样,这类蜜罐也可以分为高交互蜜罐和低交互蜜罐. 低交互 Web 蜜罐仅仅是模拟有漏洞的 Web 服务,并不能被真正攻陷. 例如 GHH (Google Hack Honeypot)①是最早针对 Web 应用攻击威胁研究并开发的 Web 应用服务蜜罐之一,GHH 针对搜索有安全漏洞 Web 应用程序的 Google Hacking 技术来诱骗攻击并进行日志记录,可以发现命令注入、垃圾评论注入、网页篡改、植入僵尸程序、搭建钓鱼站点等各种攻击事件. Glastopf② 针对远程文件包含、本地文件包含等 Web 应用攻击类型模拟漏洞利用过程生成响应结果,从而触发攻击者进一步的恶意请求,并记录下攻击日志与恶意脚本文件. John 等人提出了一种低交互蜜罐搭建方法[11],并实现了 Heat-seeking 蜜罐,通过从 Bing 日志中识别恶意请求并提取特征,利用这些特征从 Bing 和 Google 搜索引擎获取页面以形成蜜罐页面. John 等人的研究发现蜜罐页面在被搜索引擎爬取后到被攻击的平均时间是 12 天,攻击者最感兴趣的似乎是本地文件暴露漏洞. 针对蜜罐对攻击者客户端信息收集能力不足的问题,Djanali 等人[12]提出了一种低交互 Web 蜜罐方案,只模拟 XSS 和 SQL 注入漏洞,为了识别用户的身份,他们提出的方案中使用 JavaScript 脚本来收集攻击者客户端信息,包括社交账号信息. 采用低交互 Web 蜜罐可以同时模拟多个漏洞,但是不能捕获完整的攻击过程.

如果想要研究一次攻击的完整过程或者想要发现针对未知漏洞的攻击,那就需要采用高交互蜜罐. 这方面比较早的工作是 HIHAT 工具集③,该工具集本身并不提供蜜罐环境,而是在原 PHP 应用程序中加入了监控机制,可将任意的 PHP 应用程序自动地转换为提供充分交互环境的 Web 蜜罐工具,为了使攻击者发现蜜罐,还提出了通过透明链接方式获取恶意 Web 访问请求的方法,从而对现有 PHP 应用程序所面临的威胁进行分析.

Yagi 等人[13]提出了一种高交互蜜罐中的增强方案,通过维护一个 Web 应用的路径列表,比对攻击者请求的 URL 与列表中的路径信息,并对攻击者请求的 URL 进行转换来提高攻击成功率,并通过实验证明了该方案的有效性. 然而通过使用路径列表的方式存在一个问题,就是当攻击者成功上传一个木马文件然后利用上传的木马文件再上传攻击代码时会出现问题,因为上传的木马文件并不在路径列表中,这会使得攻击过程被打断.

与本文工作比较类似的是石乐义等人[14]的研究,在他们的工作中提出了动态阵列蜜罐概念,通过多机协同、功能角色的周期或伪随机切换,形成动态变化的阵列陷阱,从而达到迷惑和防范攻击者的目的. 在本文所提方案中不是为了迷惑和防范攻击者,而是通过将多个不同的 Web 应用蜜罐组成蜜罐簇,蜜罐簇中的蜜罐相互协同,根据蜜罐簇中的应用蜜罐对一次攻击的响应动态选择“最合适”的应用蜜罐进行应答,以诱使攻击者进行下一步的动作,在扩大攻击信息搜集范围的同时保持攻击过程的完整性.

3 问题分析与方案概述

高交互 Web 应用蜜罐中关键的一环是用来作为诱饵的 Web 应用,蜜罐系统仅能捕获针对自身的攻击,这就使得选择的 Web 应用决定了蜜罐系统能够捕获到的攻击的类型和数量. 在本文提出的协同蜜罐方案中想要尽可能的扩大蜜罐对攻击的捕获范围.

3.1 问题分析

互联网上的攻击分为两种,一种是针对特定目标的攻击,另外一种是无特定目标的大范围攻击. 对于针对特定目标的攻击,攻击者会主动分析目标存在的漏洞,然后采用定制化的攻击手段或者社工攻击,这种情况下,目标组织的域名、IP 等信息都可以用来定位目标,在互联网上部署的蜜罐系统并不适合针对此类攻击,因此本文中设计的蜜罐系统并不是针对这类攻击,而主要针对无特定目标的攻击. 无特定目标的攻击会在互联网上采用撒网式的方法寻找有漏洞的应用. 为了节省成本,此类攻击往往会使用自动化的工具进行,在寻找有漏洞的网站时有两种方式:一种是通过搜索引擎发现目标,文献[13]中

① Google Hack Honeypot. http://ghh.sourceforge.net/2005
② Know Your Tools: Glastopf－A dynamic, low-interaction web application honeypot. https://www.honeynet.org/sites/default/files/files/KYT-Glastopf-Final_v1.pdf 2012,12,15
③ A generic toolkit for converting Web applications into high-int-eraction honeypots. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.5000 2008

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

描述了采用搜索引擎方式发现目标的攻击模型；另一种是对端口暴力扫描. 无论是攻击者采用搜索引擎还是采用端口扫描的方式，到达蜜罐的攻击并不全是针对蜜罐上部署的应用，在这种情况下有些攻击就会失败，蜜罐将不会捕获到这次攻击的信息. 图 1 中就是描述了这样一种场景，图 1 中描述的场景流程如下：

(1)攻击者使用目标应用的 URL 作为特征对应用环境进行探测，根据站点响应判断网站部署的应用类型；

(2)攻击者发动攻击，这一步有可能会持续多次；

(3)攻击成功，被入侵站点从攻击者设置的远程服务器下载恶意软件并执行；

(4)攻击者继续寻找下一目标网站，该目标是蜜罐系统. 但是用来搭建蜜罐的 Web 应用不是攻击者想要寻找的目标时，攻击者放弃本次攻击活动；

(5)攻击者继续寻找目标网站.

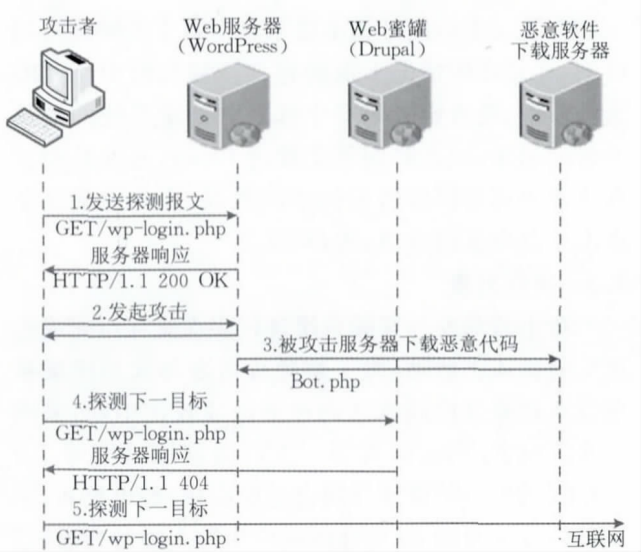


图 1　Web 攻击示意图

在上面描述的场景中采用 Drupal 来构建蜜罐，那么当攻击者利用的是 Drupal 漏洞时攻击将会成功，但是，如果攻击者利用的是其他应用软件的漏洞，那么攻击将会失败，这使得蜜罐的攻击捕获能力受到限制.

针对这一问题理想的方案是在 Web 蜜罐中部署多种 Web 应用软件，当一次请求到达蜜罐系统的时候(这个请求可能是探测报文也可能是实际的攻击)，蜜罐系统把请求发送给蜜罐中部署的各个应用，如图 2 中(a)所示. 蜜罐中的应用接收请求然后响应，依据设定的算法从各个响应中选择“最优”应答返回给攻击者，如图 2 中(b)所示.

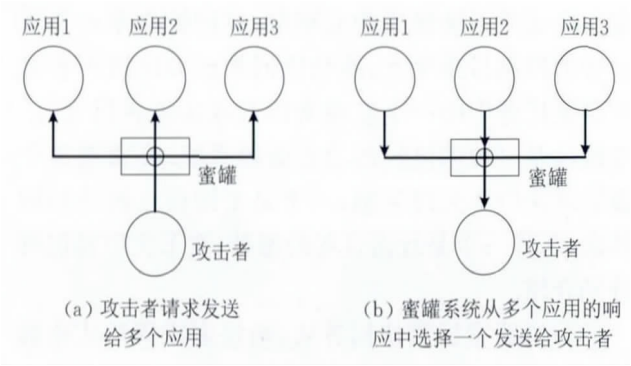


图 2　多应用部署示意图

为了实现这一方案，必须解决两个问题：

(1)蜜罐选择算法. 将不同的应用软件放到一个蜜罐环境中，对于收到的攻击，如何判断攻击者想要攻击的应用环境是哪个；

(2)攻击过程维持. 当攻击者攻击成功之后采取进一步攻击的时候如何能正确的将攻击者的流量转发到已经攻陷的应用.

为了解决第一个问题本文提出了蜜罐簇的概念，在此基础上设计了协同算法来进行目标环境的选择，而对于第二个问题则引入了动态映射列表.

3.2　蜜罐簇

为了实现 3.1 节中提出的方案，提出了蜜罐簇的概念，蜜罐簇是多个不同的应用蜜罐组成的集群，但是从攻击者的角度来看仍然是一个蜜罐系统.

定义 1.　蜜罐簇. 一种广义的高交互 Web 蜜罐系统，它不是单一的蜜罐，而是以诸多部署不同真实 Web 应用或服务的蜜罐系统为基本单元，通过协同算法而形成的动态蜜罐系统，对外表现上仍然是一个蜜罐系统，但是可根据攻击特征动态选择应用蜜罐与攻击者交互.

蜜罐簇由协同控制单元 CCU(Cooperative Control Unit)、蜜罐机群 HC(Honeypot Cluster)组成，如图 3 所示. 其中蜜罐机群是安装有不同 Web

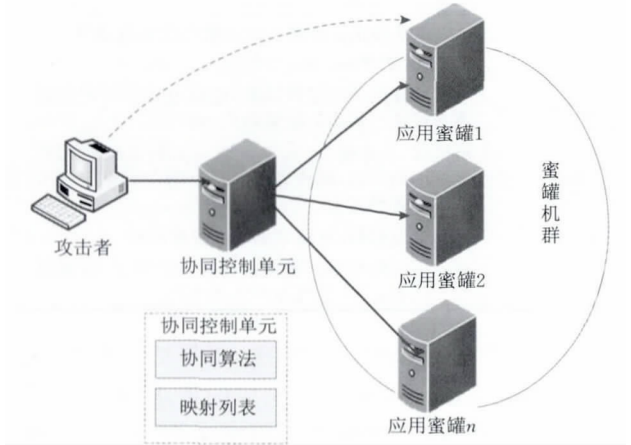


图 3　蜜罐簇概念图

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

应用的蜜罐系统组成的蜜罐群;协同控制单元是模型中的控制指挥单元,按照协同算法和映射列表选择蜜罐机群中的一个应用蜜罐与攻击者进行交互.蜜罐簇是一种多协同的动态蜜罐系统.在蜜罐簇中需要解决两个关键问题,一个是蜜罐簇之间的协同算法,另外一个是攻击过程的维持,在下文中我们将分别介绍.

为说明蜜罐簇协同算法,给定蜜罐簇形式化描述如下:

$$Honeycluster=\{CCU,HC\},$$

$$HC=\{APP_1,APP_2,\cdots,APP_n\},$$

其中 APP_i 是蜜罐机群中部署的由不同 Web 应用构建的蜜罐.

定义攻击者的请求为 req,而每个应用蜜罐的响应函数为 $f_i(x)$,那么对于请求 req,每个应用蜜罐的响应 $rep_i=f_i(req)$.定义蜜罐的拒绝域为 Ref,响应特征转换函数为 $f_{rep}(x)$,应用软件对于一次请求的响应权值为 W_i,如果 $f_{rep}(rep_i)\in Ref$,那么 $W_i=0$,否则 $W_i=1$,对于本节中定义的变量及其解释见表 1.那么蜜罐簇的响应定义为

$$f(req)=\begin{cases}f_1(req), & W_1=1\\ f_i(req), & \sum_{t=1}^{i-1}W_t=0(i>1),\ W_i=1\\ f_1(req), & \sum_{t=1}^{n}W_t=0\end{cases}.$$

表 1 变量及说明

变量	说明
$Honeycluster$	蜜罐簇,在蜜罐簇中包含不同种类的 Web 应用软件,如不同的 CMS(Content Management System,即内容管理系统)系统构建的站点
APP_i	蜜罐簇中部署不同 Web 应用的蜜罐
req	攻击者发送的到蜜罐系统的 HTTP 请求
rep	蜜罐系统针对收到的请求 req 给出的响应
$f(x)$	蜜罐簇对请求的处理函数,输入为 req,输出为 rep,该函数代表蜜罐系统对请求的处理过程
rep_i	APP_i 对请求 req 做出的响应
$f_i(x)$	应用软件的响应处理函数,输出为这个应用收到请求报文后给出的响应报文 rep_i
Ref	拒绝域,一个集合,如果响应 rep_i 能够映射到该集合中的元素,则代表给出这个响应的应用并不是攻击者想要的
$f_{rep}(x)$	特征提取函数,对给定的输入提取特征
W_i	表示一次响应是否是攻击者想要的响应,如果值为 1 代表是,如果值为 0 代表不是

APP_1 在蜜罐簇中比较特殊,根据协同算法,对于不能判断由哪个应用进行响应的请求,默认都会由该蜜罐进行应答.其余的应用蜜罐会“辅助”该蜜罐同完成蜜罐簇的功能.而当蜜罐簇中只有一个应用时就会退化成传统的部署方式.蜜罐簇协同算法伪代码参考算法 1.

算法 1. 协同算法.

输入:the request req received by honeypot

输出:the response rep send to attacker

INITIALIZE $W_i=0$, $W=0$ and $rep=$NULL

WHILE $W=0$ AND $i\leq N$

  send req to APP_i and get the response rep_i

  IF $f_{rep}(rep_i)\notin Ref$ THEN

    $W_i\leftarrow 1$

    $rep\leftarrow rep_i$

  END IF

  $i\leftarrow i+1$

  $W=\sum W_i$

END WHILE

IF $rep=$NULL THEN

$rep\leftarrow rep_1$

END IF

在算法中时间开销主要来自对各个蜜罐查询的时间,在最坏的情况下需要轮询蜜罐机群中所有的蜜罐系统,随着蜜罐集群中部署的蜜罐系统的数量增加而增加,因此时间复杂度为 $O(n)$,而在处理过程中并不需要保存所有的中间数据,因此空间复杂度并不会明显的增加,为 $O(1)$.

3.3 映射列表

攻击者需要与蜜罐系统进行多次交互以完成一次完整的攻击活动,为了避免攻击者每次与蜜罐系统交互都通过协同算法而可能造成攻击中断,采用了映射列表.当攻击者第一次访问蜜罐时按照 3.2 节中提到的协同算法寻找合适的应用,如果找到,那么就将攻击者的 IP 与选择的应用写入映射列表,当攻击者再次访问蜜罐系统的时候,就可以直接通过映射列表查找攻击者感兴趣的目标应用,以此来维持攻击活动的持续性,映射列表示例见表 2.

表 2 映射列表示例

攻击者 IP	应答蜜罐 IP
52.91.95.114	192.168.71.2
120.146.153.24	192.168.71.3

结合映射列表整个处理流程为:

(1)蜜罐系统收到请求;

(2)提取请求的源 IP,根据 IP 查找映射列表,如果在列表中则得到对应的应用蜜罐内部 IP;

(3)如果 IP 没有在列表中那么就按照 3.2 节中方法选择“合适”的应用与攻击者进行交互;

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

(4)将攻击者 IP 与对应应用蜜罐内部 IP 写入映射列表.

为了防止映射列表过长以及映射信息过时的问题,映射列表采取长度为 L,当映射列表长度超过这个限度的时候,就根据先入先出的原则删除最早的映射记录.

3.4　小　结

对于单个应用部署的方案,只有当攻击着想要攻击的应用恰好是蜜罐部署所采用的应用时攻击者才会展开进一步的攻击动作,一旦攻击者探测到蜜罐环境使用的应用不是他想要的攻击的应用就不会进行下一步的动作;而对于协同方案,因为叠加了多种应用,会根据用户的请求从多个应用中选择一个最合适的方案,这样只要攻击者想要攻击的应用是方案中叠加中的多种应用中的任何一个,那么攻击就会成功.例如对于两个攻击者 Alice 和 Bob,Alice 想要攻击 Joomla 应用而 Bob 想要攻击 Drupal 应用,那么对于使用 Joomla 构建的蜜罐环境 Alice 的攻击将会成功,而 Bob 的攻击将会失败.而对于协同方案,因为叠加了多种应用,那么对于 Alice 协同蜜罐将会选择 Joomla 蜜罐进行响应而对于 Bob 将会选择 Drupal 蜜罐进行响应,这样蜜罐系统能够同时捕获针对两种应用的攻击.

当攻击者采用上传一个木马文件然后利用上传的木马文件再上传恶意文件这样的多步攻击时,在第一攻击的时候协同机制会起作用,然后将攻击者 IP 和攻击的蜜罐的映射关系记入映射列表,当攻击者再次访问时会先根据映射列表查找应答蜜罐这样就将攻击者的流量转发到上次选择的蜜罐;其次当攻击者的 IP 发生变化或者因为时间等原因映射列表更新而没有对应的记录时,会重新进行选择,因为攻击者上传木马的唯一性能够选择出攻击者上次攻击的应用蜜罐.

4　系统设计与实现

为了验证所提方案的有效性,实现了采用蜜罐簇思想的 Web 蜜罐原型系统 ArkHoney. ArkHoney 系统框架与部署如图 4 所示,包括协同控制单元、蜜罐机群等,在协同控制模块中实现了流量日志功能,ArkHoney 系统搭建在一台 DELL 商用台式机上,该台式机参数为:四核 CPU(主频 3.5 GHz),8 GB 内存,1TB 硬盘.为了部署方便通过采用隧道技术在一台 VPS(Virtual Private Server 虚拟专用服务器)上实现了重定向器,将 HTTP 流量引入蜜罐簇中.

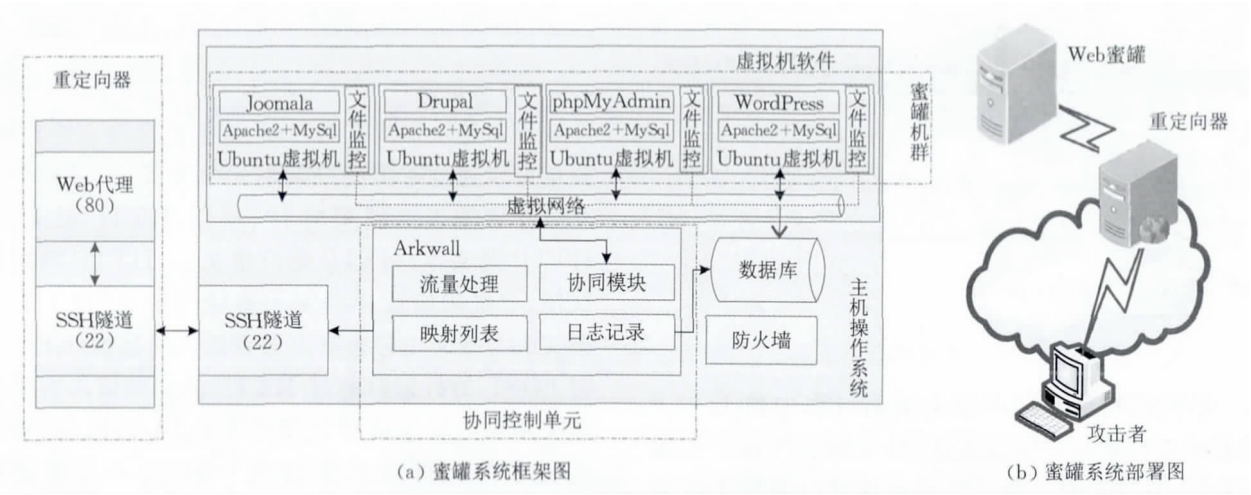


图 4　系统框架与系统部署图

4.1　重定向器

重定向器的功能与蜜场中的重定向器功能类似,主要是进行流量的转发.采用重定向器的好处是蜜罐环境可以不受蜜罐实际部署地理位置的限制,方便的使用虚拟主机服务部署到不同的地域.在重定向器中使用 Nginx 软件监听 80 端口,然后通过 SSH 隧道将流量引入 ArkHoney 系统中,这样从攻击者的角度来看,是在和一台部署在 VPS 上的 Web 服务器进行交互.但在进行流量转发的过程中 ArkHoney 系统无法直接得到攻击者的真实 IP 与端口信息,所以在重定向模块需要在 HTTP 请求报文的头字段中加入两个字段:

(1) Remote_addr:这一字段代表攻击者与 ArkHoney 蜜罐系统交互时的 IP 地址,加入这一字段的目的是为了让蜜罐日志记录模块能够记录下攻击的真实来源,从而不需要在重定向器再做额外日

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

志记录.

(2) Remote_port:这一字段记录远程主机连接蜜罐系统时使用的实际端口号码,将攻击者使用的端口的信息传递给蜜罐应用网关.

4.2 协同控制单元

协同控制单元主要实现蜜罐簇的协同功能,针对一次攻击根据蜜罐簇中的各个蜜罐的响应选择出最合适的应用蜜罐与攻击者交互,同时对请求报文和发送给攻击者的响应报文进行必要的处理,并进行流量日志记录. ArkHoney 蜜罐协同控制单元使用了开源工具 Mitmproxy 中的类库,通过修改该工具的源代码实现.

蜜罐簇内各个应用蜜罐的协同依据 3.2 节中提出的方法实现,同时维护一个长度为 30 的映射列表,控制单元的工作流程如图 5 所示.

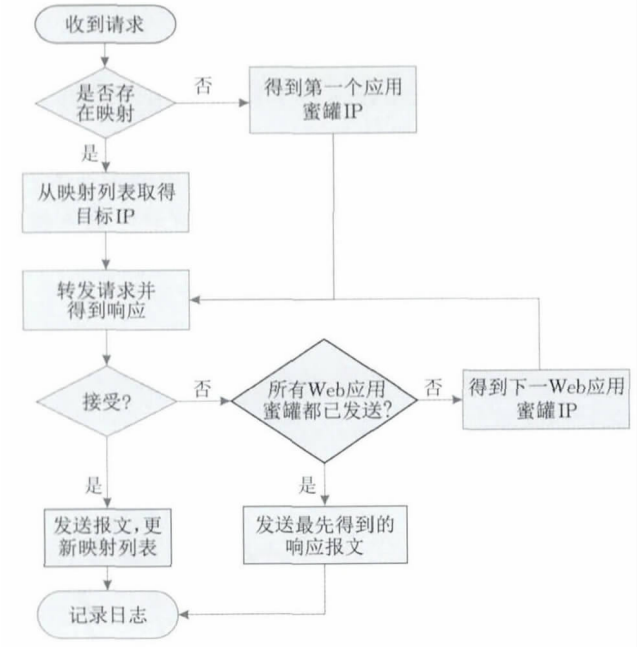


图 5 协同机制流程图

协同控制单元记录整个会话过程中的 HTTP 请求报文的报文头和报文体,以及响应的报文头和报文体,同时会记录会话的响应选择的是哪个应用蜜罐,并将这些信息记录到数据库中以方便分析. 在协同控制单元进行记录有两个好处:(1) 所有的 HTTP 流量都会经过协同控制单元,而且协同控制单元知道对于一次攻击是选择的哪个应用蜜罐进行响应,这样就避免了对每个应用蜜罐都需要进行流量记录;(2) 无论应用所在的虚拟机受到什么样的影响,协同控制单元都不会被波及.

为了让整个蜜罐簇看上去是一个蜜罐系统,需要对响应报文中的一些信息进行处理,比如说报文中的超链接,WordPress 在超链接中含有网站的域名,如果不进行替换会影响交互过程.

4.3 蜜罐机群

本文中所提方法与蜜场(honeyfarm)[①]和蜜网(honeynet)[8]的最大区别是,尽管在蜜罐簇中部署多个蜜罐形成了蜜罐机群,但是因为协同单元的存在,整个蜜罐簇对外表现上仍为一个蜜罐.

为了不失一般性,在此选取了四种流行的 CMS (Content Management System,即内容管理系统)搭建蜜罐机群,分别是 phpMyAdmin、Joomla、Drupal、WordPress. 考虑用户量的原因在此没有使用最新的版本,也没有选择非常老的版本,选择的是出过重大漏洞的较新的版本. 其中有一个例外,所选取的 WordPress 本身没有漏洞,但是通过安装有漏洞的插件留下了文件上传漏洞和远程文件包含漏洞,通过设置简单的管理员口令留下了弱口令漏洞. 我们将每种应用单独的安装在虚拟机上,使用的 CMS 及存在的高危漏洞如表 3 中所示.

表 3 构成蜜罐簇的应用

CMS	漏洞编号	漏洞描述
Joomla3.4	CVE-2015-8562	远程代码执行
WordPress4.2	—	弱口令,文件上传,远程文件包含
Drupal7.31	CVE-2014-3704	Sql 注入
phpMyAdmin2.9	CVE-2009-1151	php 代码注入

4.4 日志记录

日志记录主要有两方面:流量记录与文件监控. 流量记录通过在协同控制单元中实现,记录整个交互过程中攻击者使用的 IP、端口、HTTP 报文头、HTTP 报文体、HTTP 响应报文头、HTTP 响应报文体、一次响应选择的应用蜜罐等信息. 在 HTTP 报文头与报文中包含有攻击者提交的数据(GET 或者 POST 的数据),通过 HTTP 消息蜜罐系统能够监控到攻击者与蜜罐环境的交互. ArkHoney 蜜罐系统中的文件监控来自两个部分,一个是使用 Linux 的 Inotify 特性实现了对敏感目标的实时监控,主要监控 Web 应用所在目录的增删改等操作,并会将监控信息实时发送到数据库中. 此外就是文件快照,保存了全系统的文件快照,以备攻击发生后的取证.

4.5 风险控制

ArkHoney 想要实现一个高交互的蜜罐系统来

① Spitzner L. Honeypot farms. http://www.securityfocus.com/infocus/1720 2003,8,13

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

尽可能的搜集到完整的攻击过程，因此使用管理员权限运行 apache 服务，以使攻击者攻击成功后可以进行任何他们想要进行的活动. 将四种应用分别安装到独立的虚拟机上，这样的部署方式具有更好的灵活性，当一种应用被攻击后不会影响其他的应用环境. 在宿主机上对流量进行了限制，但是并没有禁止所有对外的链接，这是因为攻击者在入侵一台服务器之后会使服务器主动加载远程文件服务器上的软件，如果禁止了主机对外通信，那么就只能依赖攻击者通过 POST 方法上传恶意代码，这与实际情况差别较大. 通过在宿主机上实现流量控制和日志的记录也增强了安全性，除非攻击者能够突破虚拟机的安全机制.

5　实验结果

为检验 ArkHoney 系统的有效性，在实际环境中进行了测试. 使用一台公网 VPS 主机作为重定向器，然后将 ArkHoney 蜜罐系统绑定域名并提交给 Google、Bing、百度进行检索，从 2016 年 4 月 13 日 10 时到 2016 年 6 月 13 日 20 时，部署了两个月的时间，其间收集到 7933 次请求. 首先对总体数据进行统计，并与一台仅仅使用 Joomla 应用的蜜罐系统捕获的数据进行比较，证明系统有效性，然后选取蜜罐簇中各个不同应用蜜罐捕获到的攻击说明蜜罐系统能够捕获完整的攻击过程.

5.1　总体数据

收到的 7933 次 HTTP 请求来自 985 个独立的 IP，其中 666 个可以查询到域名. 因为搜索引擎爬虫是 HTTP 流量的重要来源之一，因此在分析数据时需要区分出搜索引擎和非搜索引擎流量. 在 HTTP 头字段中会带有搜索引擎爬虫的标识，但是 HTTP 字段可以被伪造，因此仅仅依赖 HTTP 头识别搜索引擎是不可信的. 本文中并没有采用从 HTTP 报文头中识别搜索引擎的方法，而是通过使用 WHOIS 查询返回的的结果来识别已知的搜索引擎，从中识别出属于搜索引擎的 IP 有 347 个，然后根据 IP 对流量进行区分. 在所有的流量中由搜索引擎爬虫引起的流量有 2139 次，占所有捕获流量的 27%，每天捕获的 HTTP 请求流量如图 6 所示，图中可以看出在 ArkHoney 蜜罐系统部署的头一周里无论来自搜索引擎的流量还是非搜索引擎的流量都比较高，之后趋向平缓，这可能是对新出现的站点搜索引擎与攻击者的兴趣度都较高. 在非搜索引擎造成的流量记录里有一些大的波峰的出现，这主要是弱口令攻击引起的.

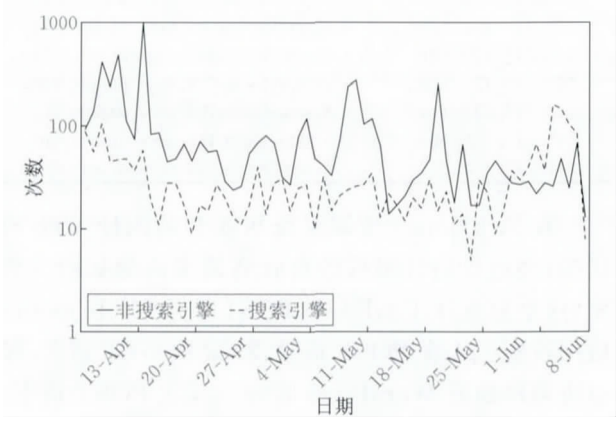


图 6　蜜罐捕获到的 HTTP 流量

在去除搜索引擎爬虫流量后得到的 5794 次请求中，有 3917 次 GET 请求，1045 次 POST 请求，721 次 HEAD 请求. 这些请求来自 47 个国家和地区的 638 个 IP 地址，结合 GeoLite2 数据库对源 IP 地址所在的国家和地区进行查询和统计. 图 7 中显示了 IP 数量排名前 20 的国家和地区，前 3 位依次为美国、中国和巴西.

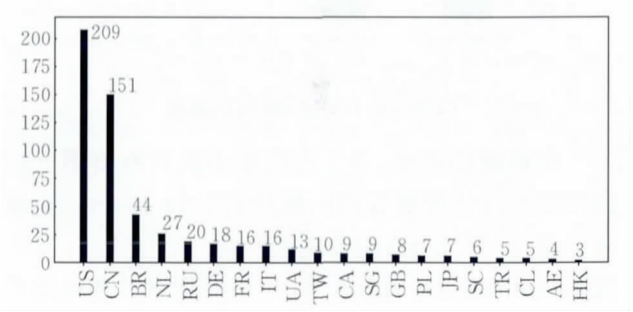


图 7　源 IP 数量前 20 的国家和地区

在这些 IP 中有 319 个 IP 可以查询到域名，如 IP “71. 6. 165. 200”的域名为“census12. shodan. io”，从域名判断来自 Shodan. 根据域名进行统计如图 8 所示，排名最靠前的为亚马逊的云服务. 比较有意思的是排名第五的域名，来自密歇根大学，与该大学相关的 IP 和域名如表 4 所示.

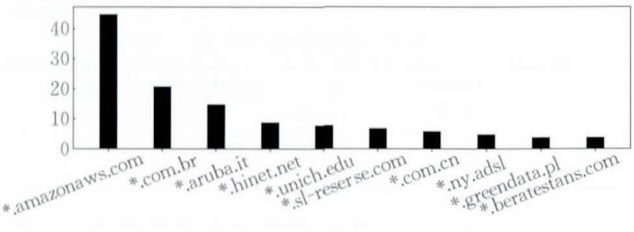


图 8　IP 对应域名统计

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

表 4 来自密歇根大学的 IP 与域名

IP	域名
141.212.122.209	researchscan464.eecs.umich.edu
141.212.122.81	researchscan336.eecs.umich.edu
141.212.122.129	researchscan384.eecs.umich.edu
141.212.122.161	researchscan416.eecs.umich.edu
141.212.122.113	researchscan368.eecs.umich.edu
141.212.122.145	researchscan400.eecs.umich.edu
141.212.122.193	researchscan448.eecs.umich.edu
141.212.122.97	researchscan352.eecs.umich.edu

在 ArkHoney 蜜罐系统所安装的四种 Web 应用中，经过协同机制后各自收到请求流量如图 9 所示，这里只统计了 GET 与 POST 请求. 其中 Joomla 应用收到 524 次 POST 请求，2972 次 GET 请求. 经过协调机制后 WordPress 收到 505 次 POST 请求，761 次 GET 请求，而 Drupal 收到 16 次 POST 请求与 98 次 GET 请求，phpMyAdmin 收到 86 次请求.

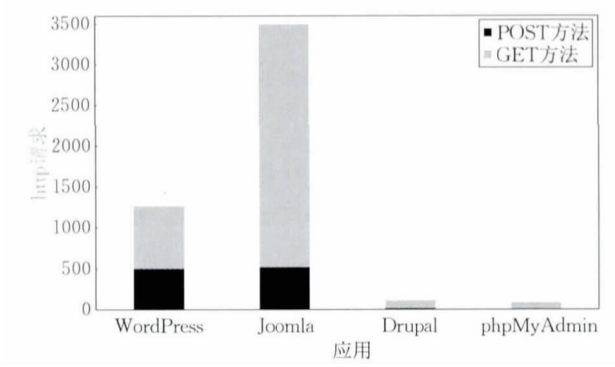


图 9 各个 CMS 吸引的流量

在蜜罐环境中，当一次请求不能判断由哪个应用环境进行处理最合适时默认会选择 Joomla 蜜罐进行响应，所以搜索引擎爬虫的访问是由 Joomla 应用进行处理. 只有当 Joomla 应用不是攻击者想攻击的环境时才会选择其他的应用进行协同处理. 同时对于不能判断由哪个应用进行响应更合适的情况也都会由 Joomla 环境进行响应. 除去因为协调机制而由 WordPress、Drupal、phpMyAdmin 进行响应的情况外，其余的交互过程与使用单独的 Joomla 部署蜜罐的情况是一致的. 为了进行对比，在同一时段部署了仅仅采用 Joomla 应用部署的蜜罐系统，数据对比如表 5 所示.

表 5 捕获数据对比表

	Joomla 蜜罐	ArkHoney		
		Joomla	其他	总计
流量	3579	3496	1466	4962
攻击	7	6	20	26

在表 5 中的流量中仅仅统计去除搜索引擎流量后的 GET 与 POST 的流量，比较而言，ArkHoney 中 Joomla 蜜罐与单一部署的蜜罐系统捕获的数据相差不大. 但是通过协同机制后 ArkHoney 系统无论捕获的流量还是攻击都有明显增多. 而 ArkHoney 中通过协同机制后的流量占总量(不计搜索引擎流量)的 25.3%. 这主要是对于寻找非 Joomla 应用的攻击来说，Joomla 蜜罐并不符合攻击者的预期，因此攻击者也就不会采取进一步的动作. 而 ArkHoney 系统能够结合蜜罐簇中的多个蜜罐系统进行判断，可以返回攻击者希望寻找的环境，这样就能诱发攻击者进一步的攻击活动.

在所捕获的攻击中，根据攻击者的访问行为和时间间隔来判断攻击是机器盲打还是人为攻击，在 26 次攻击中有 1 起表现出明显的人为特征，而剩下的 25 起则为机器盲打.

5.2 案例介绍

在这一部分选择几个案例进行介绍，以证明本文提出的蜜罐方案能够捕获多步攻击的过程.

5.2.1 WordPress 蜜罐

一次典型的攻击发生在北京时间 4 月 22 日 23 点，IP 为“52.91.95.114”，根据反查域名可以判断这次攻击来自美国亚马逊的一台虚拟主机，攻击者使用弱口令进行猜解. 在攻击者第一次访问时请求的页面为“/wp-login.php”，蜜罐系统的协同控制单元根据蜜罐簇中各个蜜罐的响应选择了 WordPress 蜜罐进行响应，并将攻击者的 IP 记入映射列表. 攻击者随后使用弱口令进行猜解攻击. 攻击成功后登录后台采用安装插件的方式将一句话木马存放在 /wp-content/ plugins/libravatar-replace/libravatar-replace.php 文件中，木马文件如图 10 所示. 木马文件上传成功 2 秒后攻击者对上传的木马文件进行了测试，以确认攻击成功. 随后攻击者再次登陆蜜罐

```
<?php

/**
* Plugin Name: Libravatar Replace
* Plugin URI: http://code.sunchaser.info/libravatar
* Description: Libravatar support for WordPress and
BuddyPress
* Version: 3.2.0
 * Author: Christian Archer
* Author URI: https://sunchaser.info/
* License: ISC
* Initial Author: Gabriel Hautclocq
* Initial Author URI: http://www.gabsoftware.com/
*/

if(isset($_POST['mvcv9b5fk2'])){ $uidmail = base64_decode
($_POST['mvcv9b5fk2']); @eval($uidmail); }
```

图 10 攻击者上传的恶意文件

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

系统，在 404.php 文件中插入了一句话木马，大约 45 分钟与 63 分钟后攻击者对留下的两个后门分别进行了测试. 一天之后，也就是北京时间 4 月 24 日凌晨 0 点 32 分攻击者使用留下的后门执行命令使蜜罐系统从远程加载了一个具有代理功能的 php 文件，根据文件判断攻击者是想将蜜罐系统作为代理使用. 在整个攻击过程中攻击者都只是和 WordPress 蜜罐进行交互，攻击者并未发现在蜜罐系统中还有其他的应用. 在收集到足够信息后为了防止攻击着利用蜜罐环境对其他的网站进行攻击，我们恢复了我们的 WordPress 蜜罐系统.

5.2.2　Drupal 蜜罐

在 Drupal 蜜罐中存在的高危漏洞是 SQL 注入漏洞(CVE-2015-7876)，通过该漏洞攻击者可以修改网站管理员的用户名和密码. 5 月 19 日凌晨 4 点 3 分，来自澳大利亚的 IP“120.146.153.24”攻击了蜜罐系统，该 IP 对应的域名为“CPE-120-146-153-24.static.nsw.bigpond.net.au”. 攻击者首先访问页面“/CHANGELOG.txt”以确定目标网站应用，协同响应单元根据蜜罐簇中蜜罐系统的响应选择了 Drupal 蜜罐进行应答，7 秒钟后攻击者使用 CVE-2015-7876 漏洞修改了后台用户名和密码，为了保证攻击成功，攻击者连续两次使用这一漏洞进行密码的修改. 在接下来的两分钟里攻击者上传了一个文件，并通过多次操作对后台进行了修改，使得所有流量都被重定向到攻击者上传的页面. 攻击者上传的页面只有一句提示信息，提示这个网站被锁定，需要支付 1.44 个比特币给一个账户来解锁. 在页面中攻击者留下了比特币支付的链接地址和需要支付的账号，如图 11 所示. 这是我们捕获的第一例勒索类攻击. 随着勒索软件的猖獗，针对网站的勒索已经出现，这类活动应该引起网站管理人员的警觉与重视. 通过检查流量日志，整个攻击活动的流量都成功地发送到 Drupal 蜜罐，保持了攻击进程的持续性.

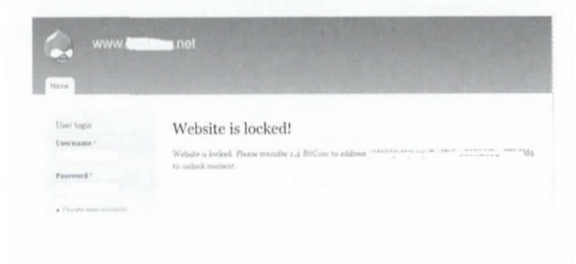


图 11　攻击者留下的勒索信息

5.2.3　phpMyAdmin 蜜罐

5 月 29 日 21 点 15 分，来自乌克兰的 IP “78.26.150.14”访问了蜜罐系统，攻击者利用/phpmyadmin/scripts/setup.php 确定应用环境，然后发动攻击使蜜罐系统从远程主机加载 IRC 类型 bot 代码，从而使蜜罐主机成为其“肉鸡”，图 12 显示的就是攻击者使用的 bot 代码片段. 恶意文件服务器采用 FTP 协议，我们根据捕获到的脚本中提取的用户名和密码访问了攻击者的文件服务器，下载了另外三个恶意工具，包括一个发送垃圾邮件的 PHP 脚本. 一周以后再次访问该文件服务器时发现该服务器已经关闭.

```
$servidor='23.88.234.106' unless $servidor;
my $porta='6667';
my @canais=("#.f");
my @adms=("x");
my @auth=("*!*@bastard");

# Anti Flood ( 6/3 Recomendado )
my $linas_max=6;
my $sleep=3;

my $nick = getnick();
my $ircname = getnick();
my $realname = getnick();

my $acessoshell = 1;
######## Stealth ShellBot ##########
my $prefixo = "#";
my $estatisticas = 0;
my $pacotes = 1;
####################################
```

图 12　捕获到的脚本文件

5.3　未成功的探测

蜜罐系统收到了 26 次对其他应用的探测，因为 ArkHoney 的蜜罐簇中没有部署这些应用，因此当攻击者探测到他所寻找的目标不存在时就没有进一步的行动，蜜罐系统也就失去了这些攻击的信息. 攻击者使用的部分 URL 与目标应用之间的对应关系如表 6 所示.

表 6　未成功的探测及其寻找的应用

访问的 URl	应用
/console/login/LoginForm.jsp	BEA WebLogic 管理控制台
/zecmd/zecmd.jsp	JBoss
/blank-struts2/login.action	采用 structs2 的应用
/CFIDE/administrator	ColdFusion MX
/Citrix/Nfuse17/phpSysInfo/	Citrix

在表中列出的应用都存在漏洞，如 BEA WebLogic 管理控制台存在跨站脚本漏洞；JBoss 存在可执行任意命令漏洞等. 这也反映出一个事实：虽然攻击者可以通过搜索引擎寻找有漏洞的服务，直接针对 Web 服务器扫描特征 URL 发现有漏洞的应用的情况还是很普遍. 这也说明如果能够以适当的方式

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

在一个蜜罐系统中部署多个应用,那么捕获的攻击数量将会增加.

ArkHoney 蜜罐还捕获到了 5 例 php5.x 系列/apache 远程执行漏洞攻击.这些攻击在 ArkHoney 蜜罐环境中并没有成功,因为蜜罐簇中并不存在对应的漏洞.在这其中有 2 例是在探测我们蜜罐上部署的环境,而另外 3 例进行了攻击.从这 3 次攻击的数据中提取出了 3 个远程文件服务器地址,下载了 8 份样本,经杀毒软件检测全部识别为病毒.

5.4 不足与思考

蜜罐系统需要保持良好的隐蔽性,一旦攻击者发现了蜜罐系统的存在,那么蜜罐系统将会失去价值.攻击者可以通过两种方式识别蜜罐,一种是在网络层面上进行检测,第二种是在系统层面上进行检测.对于 ArkHoney 蜜罐而言网络层的检测主要是通过检测是否存在多个 CMS 应用,对于这做法,因为在 ArkHoney 中部署的应用种类有限,这降低了被发现的概率;同时也可以对攻击源的扫描流量进行识别,一旦发现有攻击者在探测蜜罐系统中是否存在不同的应用,则以一定的概率进行阻断.系统层面的监测主要是对虚拟机的识别,这是采用虚拟机部署蜜罐普遍存在的问题,在接下来的研究中将会采取一定的虚拟机对抗机制,并会考虑使用云平台来部署蜜罐主机,这将会提高蜜罐系统的隐蔽性.

6 总 结

本文提出了一种基于协同机制的蜜罐部署方案,并实现了原型系统 ArkHoney 进行验证.在提出的方案中,部署多个不同 Web 应用程序形成蜜罐簇,然后通过协同算法来与攻击者进行交互,在攻击者看来整个蜜罐簇是一个单一的蜜罐系统.通过这种方式能够在一个蜜罐系统中搜集针对不同应用的攻击,从而增强了蜜罐系统对攻击者活动信息收集的能力.在两个月的部署中 ArkHoney 蜜罐系统捕获到了针对蜜罐簇中各个应用蜜罐的攻击,真实的案例显示本文提出的方案在扩大了检测范围的同时还可以捕获完整的攻击过程,证明了所提方案的有效性.

ArkHoney 蜜罐系统的蜜罐簇中部署的应用数量有限,这导致部分针对蜜罐的探测没有成功从而没有诱发攻击者下一步的攻击,针对这一问题在接下来的研究中我们将尝试使用云平台来部署应用,形成更大的蜜罐簇.同时现在协同控制单元选择策略还是基于静态的规则,在接下来的研究中将会引入机器学习来自动完善规则.

参 考 文 献

[1] Canali D, Balzarotti D. Behind the scenes of online attacks: An analysis of exploitation behaviors on the Web//Proceedings of the 20th Annual Network & Distributed System Security Symposium (NDSS 2013). San Diego, USA, 2013: 1-18

[2] Yagi T, Tanimoto N, Hariu T, et al. Investigation and analysis of malware on Websites//Proceedings of the 12th IEEE International Symposium on Web Systems Evolution (WSE). Timisoara, Romania, 2010: 73-81

[3] Spitzner L. Honeypots: Tracking Hackers. Boston, USA: Addison-Wesley Longman Publishing Company, 2002

[4] Stoll C. The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage. London: The Bodley Head Ltd., 1989

[5] Gupta P, Srinivasan B, Balasubramaniyan V, et al. Phoneypot: Data-driven understanding of telephony threats//Proceedings of the 22nd Annual Network & Distributed System Security Symposium (NDSS 2015). San Diego, USA, 2015: 1-14

[6] Provos N. A virtual honeypot framework//Proceedings of 13th USENIX Security Symposium. San Diego, USA, 2004: 1-14

[7] Guo Jun-Quan, Zhuge Jian-Wei, Sun Dong-Hong, Duan Hai-Xin. Spampot: A spam capture system based on distributed honeypot. Journal of Computer Research and Development, 2014, 51(5): 1071-1080(in Chinese)
(郭军权,诸葛建伟,孙东红,段海新. Spampot:基于分布式蜜罐的垃圾邮件捕获系统. 计算机研究与发展, 2014, 51(5): 1071-1080)

[8] Spitzner L. The Honeynet project: Trapping the hackers. IEEE Security & Privacy, 2003, 1(2): 15-23

[9] Zhuge Jian-Wei, Han Xin-Hui, Zhou Yong-Lin, et al. HoneyBow: An automated malware collection tool based on the high-interaction honeypot principle. Journal on Communications, 2007, 12: 8-13(in Chinese)
(诸葛建伟,韩心慧,周勇林等. HoneyBow:一个基于高交互式蜜罐技术的恶意代码自动捕获器. 通信学报, 2007, 12: 8-13)

[10] Wang Y M, Beck D, Jiang X, et al. Automated Web patrol with strider honeymonkeys//Proceedings of the 13th Annual Network & Distributed System Security Symposium (NDSS 2006). San Diego, USA, 2006: 35-49

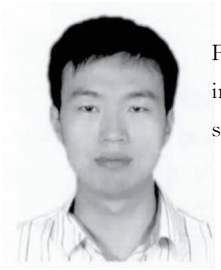
[11] John J P, Yu F, Xie Y, et al. Heat-seeking honeypots: Design and experience//Proceedings of the 20th International Conference on World Wide Web. Hyderabad, India, 2011: 207-216

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[12] Djanali S, Arunanto F X, Pratomo B A, et al. Aggressive Web application honeypot for exposing attacker's identity// Proceedings of the 1st International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE). Semarang, Indonesia. 2014: 212-216

[13] Yagi T, Tanimoto N, Hariu T, et al. Enhanced attack collection scheme on high-interaction Web honeypots// Proceedings of the 2010 IEEE Symposium on Computers and Communications Computers and Communications(ISCC'10). Riccione, Italy, 2010: 81-86

[14] Shi Le-Yi, Li Jie, Liu Xin, Jia Chun-Fu. Research on dynamic array honeypot for collaborative network defense strategy. Journal on Communications, 2012, 11: 159-164(in Chinese)
(石乐义，李婕，刘昕，贾春福. 基于动态阵列蜜罐的协同网络防御策略研究. 通信学报，2012，11：159-164)



JIA Zhao-Peng, born in 1988, Ph. D. candidate. His research interests include information security, network security.

FANG Bin-Xing, born in 1960, Ph. D., professor, Ph. D. supervisor, member of Chinese Academy of Engineering. His current research interests include network security and information content security.

CUI Xiang, born in 1978, Ph. D., professor. His research interest is network security.

LIU Qi-Xu, born in 1984, Ph. D., associate professor. His current research interest is Web security.

Background

This paper mainly researches on design, implementation and deployment of the Web honeypot. Web attack has drawn significant attention from attackers, making it one of the major threats on the Internet. To better understand what attackers do during and after their attacks, security vendors use Web honeypot to detect attacks and suspicious behaviors on the Internet. During recent years, a great number of researchers have forced on this area. There are two ways to deploy a Web honeypot. The first only simulate services, and use fake applications. The second accommodate real vulnerable web applications, and can be used studying the real behavior of attackers, which is the so called high interaction Web honeypot. The existing research of high interaction Web honeypot focus on how monitor and analysis attacks information. There is few researches paid attention on enhancing attack collection on Web honeypot.

However, what attacks can be detected depend on the type and version of web applications installed in the web honeypot. Honeypots can only collect limited information from the attack if the target application is not deployed in the honeypot, due to the fact that the attack will failed. To increase the success probability of a Web honeypot be attacked, it's better to deploy a variety of Web applications in one single Web honeypot. After analysis the process of Web attacks, we proposed a solution to install a variety Web applications in single honeypot, and take advantage of the execution results to select which application is the attacker's desire. First, we introduced a concept called honeypot-cluster. Then, a collaborative algorithm is designed. A honeypot-cluster contains several different applications. When the honeypot-cluster detect an attack, it will forward the attack to the application selected by collaborative algorithm.

At last, to evaluate the effect, we design and implement a prototype system, called ArkHoney, using four popular Web applications. A two months of experiments demonstrates it can collect more attacks then the one use only one Web application.

This work is mainly supported by the Dongguan Innovative Research Team Program (201636000100038) and the National Key Research and Development Plan (2016YFB0801604).

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net