

# 高交互客户端蜜罐 Agent 系统研究

贾志洋<sup>1</sup>, 李伟伟<sup>2</sup>, 王勇刚<sup>1</sup>, 高 炜<sup>3</sup>, 夏幼明<sup>3</sup>

(1. 云南大学旅游文化学院, 云南 丽江 674100; 2. 宁德职业技术学院, 福建 福安 355000; 3. 云南师范大学, 昆明 650040)

**摘 要:**根据国内外客户端蜜罐系统的发展现状以及恶意网页攻击机理,设计了一个基于高交互客户端蜜罐技术的协同 Agent 工作模型,并实现了 Malicious Web Site Hunter(MWSH)恶意网页检测系统。系统主要包括控制子系统、高交互级蜜罐子系统及服务子系统三个部分。系统利用 Agent 技术的协作性以及客户端蜜罐技术的准确性来发现网络中的恶意网页。利用基于 VSK 逻辑的 Agent 形式化描述语言对系统涉及的 Agent 进行了形式化描述,并做了具体实现,不仅可以描述 Agent 和它的环境信息特征,而且可以描述其完整状态变化过程。

**关键词:**客户端;智能体;蜜罐;恶意网页

**中图分类号:**TP309.5

**文献标识码:**A

## 引 言

浏览器是应用广泛的客户端应用程序,是用户使用网络的最主要平台之一。当前计算机安全领域中有很大一部分问题直接或间接来源于浏览器应用,浏览器的安全保护已经成为计算机安全的一个重要组成部分。恶意网页在全球互联网上已成蔓延之势,其比例已高达网页总量的 10% 左右<sup>[1]</sup>,虽然目前有许多浏览器安全产品,但是利用恶意网页发动的网络攻击却没有得到有效遏制,针对恶意网页攻击行为的研究也只处于起步阶段,用户计算机面临着严重的安全威胁。

国内外对恶意网页的工作机理及其防御机制都做了大量的研究。很多研究都是基于蜜罐系统的,比如微软研究院于 2005 年研制的一款基于 IE 浏览器的高交互式蜜罐系统 HoneyMonkey<sup>[2]</sup>,HoneyMonkey 由一系列装有 Windows 操作系统的处于不同安全级别的虚拟计算机组成,系统运用了黑盒分析方法,但是在效率和准确性上 HoneyMonkey 还是存在着很多问题。美国华盛顿大学于 2005 年成功开发的 Spycrawler<sup>[3]</sup>,也是具有私人版权的并且是基于高交互性以事件驱动的蜜罐系统。其研究重点是间谍软件(Spyware),故此没有涉及到其它类型的恶意代码。Kathy Wang 于 2004 年开发的一个

开源蜜罐系统 HoneyClient 系统<sup>[4]</sup>是一个高交互式蜜罐,HoneyClient 系统基于虚拟化技术,所以在被感染后,它可自动被重置为干净的状态。这种系统可与潜在的恶意服务器交互,并可以监视系统,可以在与服务器交互期间或之后查找未授权的状态改变。国内对于恶意网页的研究起步较晚,诸葛建伟等<sup>[5]</sup>对恶意网页进行了研究。他们总共分析了 145 000 个国内的网页,根据他们对恶意网页的定义,发现了 214 9 个网页含有恶意代码,这一数值占到了全部网页的 1.49%。HoneyBow 系统采用蜜网技术,同时兼容传统蜜网和虚拟蜜网技术,通过选取感染前和感染后不同时间点上的底层文件系统视图进行差异比对,以此发现网页恶意代码,系统可扩展性不高,对硬件设备要求较高,而且对恶意代码的入侵机制及其防御没有进行深入研究。孙晓妍将蜜罐系统与爬虫系统相结合,通过设计一个网络爬虫来获取 URL 数据源,利用蜜罐内的客户端引擎自动启动 IE 浏览网页,并监控通过浏览恶意网页下载的恶意软件。最后分析恶意代码,将恶意网页加入黑名单,以此来防御恶意网页<sup>[6]</sup>。不足之处在于其仅对系统进行设计,并没有深入研究恶意网页攻击机理。

## 1 系统设计

由于网页相关编写语言的简单性,大部分网页恶意

代码编写者都采用了各种各样的方法和技术来逃避反病毒软件的检测,以使它们实现入侵用户计算机的目的。所以,作为一个优秀的网页恶意代码检测系统,首要的是准确、快速的检测出恶意网页以阻止其入侵。其次,系统应该具备在网络上自由爬行网页的功能,以使其适应于大规模搜索恶意网页。最后,该系统应该具备足够的健壮性,以应对硬件故障及恶意网页入侵行为。根据这些原则,本文基于 Agent 技术与客户端蜜罐系统设计并实现了 Malicious Web Site Hunter(MWSH) 恶意网页检测系统。

### 1.1 系统结构

MWSH 系统主要由 Service Agent、Control Agent、高交互蜜罐系统(包括 Monitor Agent) 以及网络爬虫等部分构成,如图 1 所示。Control Agent 设置在控制服务器和备份控制服务器中,正常情况下,控制服务器上的 Control Agent 负责全局事务,备份控制服务器上的 Control Agent 只负责实时接收全局服务器上的备份日志信息,处于半休眠状态。当控制服务器失效,备份控制服务器上的 Control Agent 自动启动,备份控制服务器升级为控制服务器,备份控制服务器中的 Control Agent 接管全局事务。其目的主要是为了增强系统的健壮性,防止控制服务器成为整个系统的单一失效点。Service Agent 和高交互蜜罐系统分别设置于各区域服务器和检测服务器。

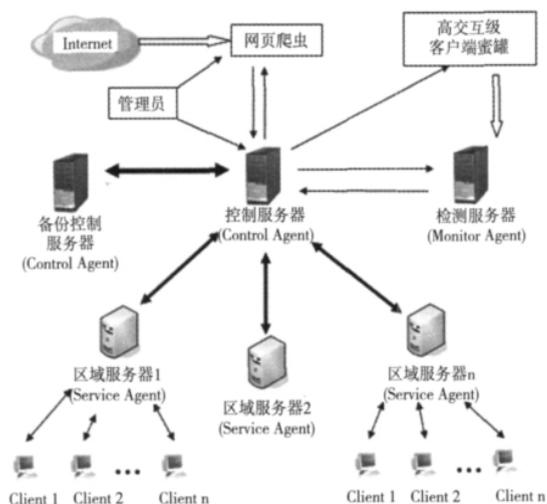


图 1 MWSH 系统的总体结构

### 1.2 系统工作流程

MWSH 系统可细化为两个流程途径,分别为(1) - (2) - (3a) - (4) - (5) - (6) 和(1) - (2) - (3b) - (4) - (5) - (6)。蜜罐系统浏览的网页地址来源可以由客户端提供也可以由网页爬虫自动获取。

(1) 客户端查询/举报 URL,提交相关信息至区域服务器。

(2) 区域服务器计算 URL 的 Hash 值,在本地数据库中查找。若数据库中有该条记录,则直接返回查询结果给客户端,若数据库中无该条记录,则计算该 URL 的优先权并将其插入待提交 URL 队列中。

(3a) 控制服务器对各区域服务器提交的查询/举报 URL 按照优先权进行排序,插入待检测 URL 队列。

(3b) 当控制服务器中待检测 URL 队列长度小于某阈值时,控制 Agent 激活网页爬虫程序,利用 URL 种子提取网页中所有 URL 链接,并赋予其优先权,作为待检测 URL。

(4) 蜜罐从控制服务器中的待检测 URL 队列获取 URL 并浏览,由监控 Agent 负责监控并记录浏览器浏览 URL 时蜜罐系统信息的改变状况。

(5) 监控 Agent 将该日志信息提交至控制服务器,由控制 Agent 分析日志并形成网页危害性报告。

(6) 控制服务器向各区域服务器下发网页危害性报告。

### 1.3 系统总体说明

MWSH 系统由三个子系统组成:控制子系统、高交互蜜罐子系统及服务子系统。

控制子系统安装于控制服务器端,由 Control Agent 具体实现,Control Agent 主要负责对整个系统进行协调控制,对其它智能 Agent、网页爬虫以及蜜罐系统等进行管理和任务调度。数据分析模块和队列模块也集成在控制子系统中。这里的数据分析是指从蜜罐系统日志数据中获取恶意网页行为相关信息的能力。队列模块则是负责为浏览器提供源 URL 相关事宜。另外,控制子系统还为服务子系统提供数据库接口以接收和反馈用户查询,最终实现集网页检测、恶意网页过滤于一体。另外,网页爬虫也处于控制子系统内。其主要负责从网络中大量提取 URL,然后将该 URL 集作为蜜罐检测样本。它是大规模检测网页,建立网页信息数据库必不可少的一部分。

高交互蜜罐子系统分为队列模块、数据分析模块和蜜罐自身。其中,队列模块和数据分析模块嵌入到控制子系统中(如前所述)。本文的客户端蜜罐(包括 IE 浏览器和 Monitor Agent) 安装于检测服务器端,运行于虚拟机环境下。在虚拟机环境中首先启动 Monitor Agent 通过其记录浏览器运行后而引发的所有事件,一旦检测到系统被未授权修改,即当蜜罐系统受到恶意网页攻击后,虚拟机在 IE 浏览器关闭后就重新恢复到原始的系统状态,以等待与下一个网页进行交互。同时负责将蜜罐子系统所有相关日志信息转储至控制服务器数据库。另外,由于蜜罐系统的特殊性,导致其有可能很轻易的被攻击者攻陷。一旦被攻击者攻陷,它就可能被用来攻击网络中的其它计算机系统。所以,它又

涉及到数据控制问题。数据控制最主要的目的就是确保攻击者在入侵蜜罐的过程中不会有意或无疑的对其其它计算机系统造成危害。为此,在发挥操作系统自身安全功能的前提下,还采用了限制连接数、带宽限制以及蜜墙等手段以监控蜜罐流出数据。蜜罐系统需要考虑的另一个问题是数据捕获。数据捕获包括对攻击者入侵蜜罐系统时的监控及日志记录。捕获数据的最终目的是为了分析攻击者入侵时运用的攻击、采用的策略及步骤。捕获过程中最困难的是如何在攻击者没有发现其被监控的状态下尽可能多的获取入侵信息。

服务子系统安装于区域服务器上,由 Service Agent 负责实现,其主要功能是接收客户端查询、举报的网页信息,返回查询结果;为控制子系统提供接口,搜集并提交未检测的 URL 以作为蜜罐检测样本。将服务子系统和控制子系统分别安装于区域服务器和控制服务器上的一个主要目的是为了减轻控制服务器的访问压力,尽可能的降低系统失效的风险。

## 2 基于 VSK 逻辑的 Agent 实现

在设计一个 Agent 时,经常需要对此 Agent 和它的环境信息特征进行描述。而且,许多任务的执行取决于 Agent 是否能够访问环境中的某些信息,如果这些信息是不可访问的,那么 Agent 就不可能完成任务。同样,为了让 Agent 顺利地完成任务,Agent 知道一些特定的信息也是很重要的,而且 Agent 的感觉器必须能够感知到这些信息。除此以外,Agent 还应该能够对来自环境的各种信息进行存储。针对上述情况,Michael Woolbridge 和 Alessio Lomuscio<sup>[7]</sup> 提出了 Agent 系统的 VSK 逻辑,给出了 VSK 逻辑的语义模型和公理体系,并对其完备性、相容性进行了证明。VSK 逻辑是对环境中的 Agent 信息特征进行推理的多模态逻辑,它引入了三个模态算子: V(Visibility) 模态算子、S(Perception) 模态算子和 K(Knowledge) 模态算子,用它们来描述 Agent 和它所处的环境的信息特征。VSK 逻辑可以表示环境中客观真实的信息。换句话说,将 Agent 可访问的信息或对 Agent 来说知道的关于环境的可知的信息视为真值,并且可以形式化地描述 Agent 从环境中感知到的信息。其中, V 模态算子、S 模态算子和 K 模态算子分别表示 Agent 可访问的信息、可感知的信息和知道的信息。

MWSH 系统中的 Agent 使用扩张的 Snetl 语言对 VSK 逻辑的语义部分进行描述,从而给出 Agent 系统语言,该语法的定义是由巴科斯范式<sup>[8-9]</sup>的形式给出。

由于 Service Agent、Control Agent 和 Monitor Agent 等模块构成 Agent 系统,考虑到效率问题,采用简单反应式 Agent 模块作为本系统的基本构件。

以 Service Agent 为例分析 MWSH 系统的通信机制, Service Agent 内核由执行机、决策模块、反应模块、邮箱、通信模块、数据库模块以及数据收集模块组成,各组成部分具体功能如下:

(1) 执行机: 负责完成消息分派,功能模块的执行控制等。

(2) 决策模块: 负责冲突检查和消解,并决定当前动作和通信的选择。

(3) 反应模块: 负责具体操作的实际执行。

(4) 邮箱: 提供 Agent 通信的临时存储区。

(5) 通信模块: 负责 Agent 和环境以及其它 Agent 的通信。

(6) 数据库模块: 负责网页信息的查询、插入和删除等功能。

(7) 数据收集模块: 负责从客户端服务请求中收集未知 URL 信息。

(8) 安全保障模块: 负责阻止恶意用户攻击。

王勇<sup>[10]</sup>在信息系统定义的基础上,根据 Agent 的特征,把三元组扩展成为五元组,给出多 Agent 信息系统的集合定义:称  $(U, S, E, R, T)$  是一个信息系统,其中  $U$  为 Agent 集合,  $U$  中的每一个元素  $Agent_i (i \leq n)$  称为一个 Agent。

$$U = \{ Agent_1, Agent_2, \dots, Agent_n \} \quad (1)$$

$S$  为  $Agent_i (i \leq n)$  内部状态属性,是  $Agent_i (i \leq n)$  的局部变量,采用集合的方法表示这些状态属性为:

$$S = \{ Internal A_1, Internal A_2, \dots, Internal A_n \} \quad (2)$$

$E$  为  $Agent_i (i \leq n)$  感知外部环境状态的属性,是感知器的输入接口参数的类型,采用集合表示为:

$$E = \{ External A_1, External A_2, \dots, External A_n \} \quad (3)$$

$R$  是  $Agent_i (i \leq n)$  返回的状态属性,是输出参数的类型,这些属性作为其他 Agent 的输入,采用集合表示为:

$$R = \{ Return A_1, Return A_2, \dots, Return A_n \} \quad (4)$$

$T$  为  $Agent_i (i \leq n)$  的任务集合,  $Agent_i$  个体具备改变外界环境和自身的能力,其不仅可以改变内部状态属性  $S$  的值,而且可以改变返回的状态属性  $R$  的值,  $F_s$  表示内部状态属性所对应的函数,  $F_r$  表示返回状态属性对应的函数,  $V_i$  表示值。

$$T = \{ Fs: Internal A_i \rightarrow V_i; Fr: Return A_i \rightarrow V_i \} \quad (5)$$

ServiceAgent 可能遇到的外部事件包括:客户端提交查询/举报信息,Control Agent 广播新数据,服务器失效等。其对应的知识规则见表 1(1 表示肯定,0 表示否定,null 表示不需要经过此操作)。

表1 Service Agent 知识表

U	S		E			R					T		
	系统可用	记录匹配	查询	举报	有效	接收数据	全局服务器更新	提供服务	拒绝服务	插入上传队列	应答	更新数据	更新服务器信
Service Agent	1	1	1	0	1	0	0	1	0	0	0	0	0
	1	1	0	1	1	0	0	1	0	0	0	0	0
	1	0	1	0	1	0	0	0	0	1	1	0	0
	1	0	0	1	1	0	0	0	0	1	1	0	0
	1	null	1	0	0	0	0	0	1	0	1	0	0
	1	null	0	1	0	0	0	0	1	0	1	0	0
	1	null	0	0	0	1	0	0	0	0	1	1	0
	1	null	0	0	0	0	1	0	0	0	1	0	1

根据表1 Service Agent 规则如下:

Rule 1: S{1,1} E{1,0,1,0,0} R{1,0,0,0} T{0,0}

Rule 2: S{1,1} E{0,1,1,0,0} R{1,0,0,0} T{0,0}

Rule 3: S{1,0} E{1,0,1,0,0} R{0,0,1,1} T{0,0}

Rule 4: S{1,0} E{0,1,1,0,0} R{0,0,1,1} T{0,0}

Rule 5: S{1, null} E{1,0,0,0,0} R{0,1,0,1} T{0,0}

Rule 6: S{1, null} E{0,1,0,0,0} R{0,1,0,1} T{0,0}

Rule 7: S{1, null} E{0,0,0,1,0} R{0,0,0,1} T{1,0}

Rule 8: S{1, null} E{0,0,0,0,1} R{0,0,0,1} T{0,1}

表2 恶意行为危害等级分布

危害等级	恶意行为
第0级	无恶意行为
第1级	1、安装 ActiveX 控件
	2、Browser Helper Object 被更改
	3、触发多于1个IE进程
第2级	1、注册表 IE 项目被更改
	2、触发未知子进程
	3、CPU 内存使用过度
第3级	1、注册表禁止运行项被更改
	2、触发多于5个IE进程
第4级	1、注册表自动运行项被更改
	2、文件系统特定文件被更改(包括 .exe , .com , .dll , .bat 等文件)
	3、触发已知恶意进程

表3 实验结果统计

类型	1级	2级	3级	4级	0级	总计
电影	6	3	2	4	474	489
软件	3	2	5	2	517	529
音乐	3	4	1	10	524	542
成人	8	3	17	24	369	421
游戏	2	1	5	7	961	976
网购	0	2	8	4	659	673
合计	22	15	38	51	3504	3630

### 3 实验与结果分析

由于高交互级客户端蜜罐受自身特点及硬件设备的制约,以致无法在本质上直接提升检测网页的速度。本文假定网络上的网页数量趋于无限,而检测资源是有限的,而且检测的最终目的是发现尽可能多的恶意网页,并在普通用户浏览前对其过滤。为此,本文提出了一种对待检测 URL 进行预处理的思想,即根据优先权对 URL 进行排序,优先权高的 URL 先被蜜罐检测,优先权低的 URL 后被蜜罐检测。

系统通过监控网页运行时的 CPU 使用量、内存使用量、进程、文件系统以及注册表系统来识别恶意网页。网页危害性分为五个等级,包括第0级、第1级、第2级、第3级和第4级,级别越高,越危险。其中,第0级表示该网页无恶意行为,第4级表示危害程度最高。同一网页具有多种恶意行为时,以最高危害行为评价其危害性(表2)。

选取“电影”,“软件”,“音乐”,“成人”,“游戏”,“网上购物”共6个常用关键字,分别利用 Google 搜索引擎,提取搜索结果第1页记录作为爬虫种子节点。MWSH 系统共提取 3630 个有效网址并实际运行检测,其结果统计分析见表3。

根据表3可以得出网络中恶意网页约为 3.45%,由于本文采用的网页种子是 Google 搜索结果页第1页,而 Google 也对恶意网页进行了初步的过滤<sup>[11]</sup>,故认为网络中实际存在的恶意网页比例会略大于本文检测结果。但是,本文仍真实的反应了目前搜索引擎提供的搜索结果网页的安全状态。很多恶意网页使用隐蔽后台下载操作,在客户计算机上植入病毒、插件以达到不可告人的目的。“成人”类型网页中,恶意网页比例依然最为严重,达到 12.35%。

### 4 结束语

浏览器攻击是近几年才出现的问题,目前对其研究仍处于起步阶段,本文也只是对其粗略探讨。由于网络攻击技术日益更新,以至于不得不更新知识以改良目前的恶意网页检测方法,故可以考虑构造一个具备自学习能力的混合式蜜罐系统,通过设计一个反馈回路利用知

识更新模型获取新知识,以摆脱简单的依靠非授权修改来识别恶意网页,以提高准确性。

#### 参考文献:

- [1] Niels Provos, Dean McNamee, Panayiotis Mavrommatis. The Ghost In The Browser Analysis of Web-based Malware [C]. Proceedings of HotBots, Cambridge [S. l.]: [s. n.]: 2007.
- [2] Wang Y-M, Beck D, Jian X X, et al. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities [C]. Proceedings of the 13th Network and Distributed System Security Symposium. San Diego, California: [s. n.]: 2006.
- [3] Moshchuka, Bragnt, Grbblesd, et al. A crawler-based study of spyware on the web [EB/OL]. [2009-4-05]. <http://www.cs.washington.edu/homes/gribble/papers/spycrawler.pdf>
- [4] Wang G K. Using honeyclients to detect new attacks [EB/OL]. [2009-04-05]. <http://www.synacklabs.net/honeyclient/Wang-Honeyclients-RECON.pdf>.
- [5] 诸葛建伟, 韩心慧. 一个基于高交互式蜜罐技术的恶意代码自动捕获器 [J]. 通信学报, 2007, 28 (12): 8-13.
- [6] 孙晓妍, 王洋, 祝跃飞, 等. 基于客户端蜜罐的恶意网页检测系统的设计与实现 [J]. 计算机应用, 2007, 27(7): 1613-1615.
- [7] Wooldridge M, Lomuscio A. A Logic of Visibility, Perception, and Knowledge: Completeness and Correspondence Results [C]. Proceedings of the Third International Conference on Pure and Applied Practical Reasoning. London UK: [s. n.]: 2000.
- [8] 夏幼明, 刘海庆, 徐天伟. 基于语义网络的知识表示的形式转换及推理 [J]. 武汉大学学报: 信息科学版, 2001, 26(4): 369-373.
- [9] 夏幼明. 语义网络的知识获取及转换的研究 [J]. 云南师范大学学报, 1999, 19(6): 40-44.
- [10] 王勇, 黄国兴. 多 Agent 信息系统知识发现研究 [J]. 计算机科学, 2008, 35(1): 184-186.
- [11] 唐子蛟, 李红婵. 基于 PageRank 算法的商业网站推广策略研究 [J]. 四川理工学院学报: 自然科学版, 2009, 22(6): 54-56.

### Research of Agent-based Client Honey-pot System

JIA Zhi-yang<sup>1</sup>, LI Wei-wei<sup>2</sup>, WANG Yong-Gang<sup>1</sup>, GAO Wei<sup>3</sup>, XIA You-ming<sup>3</sup>

(1. Tourism and Literature College of Yunnan University, Lijiang 674100, China;

2. Ningde Vocational and Technical College, Fuan 355000, China;

3. Yunnan Normal University, Kunming 650040, China)

**Abstract:** According to research of client honey-pot system and mechanism of the malicious web sites attacks, a highly interactive client honey-pot system called Malicious Web Site Hunter was designed and implemented based on Agent working model. The system is composed of control sub-system, high-interaction client honey pot sub-system and service sub-system. It is a malicious websites detecting with intelligence, which make full use of Agent's cooperation and honey pot's accuracy. The formal description of Agent of the system is implemented based on Agent VSK Logic formal description language, which not only to describe the Agent and its characteristics of environmental information, but also can describe the complete state of change process.

**Key words:** client; Agent; honey-pot; malicious web pages