网络设备协同联动模型

臧天宁[1] 云晓春[1),2),3)] 张永铮[2)] 门朝光[1)] 孙建亮[2)]

[1)](哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

[2)](中国科学院计算技术研究所 北京 100190)

[3)](国家计算机网络应急技术处理协调中心 北京 100029)

摘 要 在开放的互联网环境下,大规模分布式网络恶意行为日益增多.发生在不同地理位置、不同时间段的安全事件可能存在潜在的隐藏关系.作者基于通用图灵机思想,提出了一个处理大规模网络安全事件的协同联动模型(Coordinative Running Model,CRM).在形式定义的基础上,从人机交互角度分析模型层次结构,由不同部件构建模型系统结构,并实现了面向基础网络的协同联动系统(Coordinative Running System,CRS),且与基于安全域的安全操作中心(Security Operating System,SOC)模型进行了对比分析.在僵尸网络的检测和追踪、DDoS 攻击事件关联以及僵尸网络与 DDoS 攻击源关系分析三个应用实例中,CRS 协调骨干网上不同类型安全设备共同工作.典型数据的分析结果表明,CRS 为分析不同时间及不同空间安全事件之间关系,挖掘各事件关联后的更深层次安全隐患提供了有力平台.

关键词 网络安全;协同联动;图灵机;DDoS;僵尸网络

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2011.00216

A Model of Network Device Coordinative Run

ZANG Tian-Ning[1] YUN Xiao-Chun[1),2),3)]

ZHANG Yong-Zheng[2)] MEN Chao-Guang[1)] SUN Jian-Liang[2)]

[1)](*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001)

[2)](*Institute of Computing and Technology, Chinese Academy of Sciences, Beijing* 100190)

[3)](*National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing* 100029)

Abstract Internet is an open network environment, large-scale distributed malicious behaviors is increasing day by day on the internet. Potential relationships may exist among network security incidents which occur at different positions and time. In order to deal with those troubles, this paper presents a Coordinative Running Model (CRM) based on Universal Turing Machine. Formal mathematical definition of the model is proposed. Architecture of the model is hierarchy, and the model consists of several important components, which include storage component, interface system and coordinative running engine etc. On the basis of the above work, a Collaborative Running System (CRS) is implemented for analyzing distributed incidents of backbone network. Furthermore, this model is compared with the Security Operation Center (SOC). For three application scenarios, namely botnet tracking, correlation analysis for alerts of Distributed Denial-of-Service (DDoS) attack and relationship analysis between DDoS attack source and botnet, different types of monitoring devices of the backbone network work together through CRS. The analy-

收稿日期:2010-04-26;最终修改稿收到日期:2010-12-17.本课题得到国家自然科学基金(60703021,60873138)、国家"八六三"高技术研究发展计划项目基金(2007AA01Z444,2007AA01Z467,2007AA01Z474,2007AA010501)、博士后科研启动金(LBH-Q08124)资助.
臧天宁,男,1978 年生,博士研究生,主要研究方向为僵尸网络、协同分析.E-mail: tn.zang@gmail.com. **云晓春**,男,1971 年生,博士,教授,博士生导师,主要研究领域为网络安全、互联网建模和网络测量. **张永铮**,男,1978 年生,博士,副教授,主要研究方向为网络安全和网络安全评估. **门朝光**,男,1963 年生,博士,教授,博士生导师,主要研究领域为可信计算等. **孙建亮**,男,1984 年生,硕士研究生,主要研究方向为协同分析.

sis results of typical security incidents data show that CRS is efficient and effective to collaboratively analyze the relations of large-scale security incidents at different time and space, and CRS is a powerful platform for analyzing hidden danger among different incidents.

Keywords　network security; coordinative running; turing machine; DDoS; botnet

1 引　言

随着计算机通信技术的进步，大规模网络恶意行为日趋复杂和多样，越来越多的研究工作关注多网络安全设备的融合和协同以应对大规模分布式网络安全事件. 美国加州大学Davis分校在20世纪90年代就研发了DIDS(Distributed Intrusion Detection System)系统[1]，把分散部署的若干个网络监视器和主机监视器采集的信息送到中心节点DIDS Director，集中分析处理. 德克萨斯农工大学提出了对等体组织结构的CSM(Cooperating Security Managers)[2]系统，由若干个基于主机的入侵检测和响应系统CSM组成，设立攻击可疑度作为响应因素. 美国国防部高级研究计划局(DARPA)资助的EMERALD[3]项目构建了一个集成误用检测和异常检测的大型分布式入侵检测与分析系统. 日本信息化推进机构(Information technology Promotion Agency，IPA)开发的入侵检测代理系统(Intrusion Detection Agent system)[4]采用两层架构，应用移动代理技术自动收集信息. 法国MIRADOR项目提出了网络恶意行为的协同识别模型(Cooperation and Recognition of Malevolent Intentions，CRIM)[5-6].

这些研究工作主要是针对分布式入侵检测系统，通过告警聚合去除冗余告警，降低虚警率，然后进一步关联告警信息，识别入侵行为，分析入侵意图.

Renaud[7]描述的安全操作中心(Security Operation Center，SOC)结构模型，融合了网络管理和安全管理. 首先根据大量网络恶意行为的模式关联，建立入侵行为模板，存储于知识库. 然后按照时序模式匹配等方式处理各种网络安全监测设备探测到的异常信息，监测网络安全事件. 并根据系统漏洞和脆弱性分析，建立系统安全策略模板. Fortinet公司提出了一种七层安全技术，统一威胁管理(United Threat Management，UTM). 将多种安全特性集成于一个硬设备里，构成统一管理平台. 具备网络防火墙、网络入侵防御和网关防病毒等功能. 是一种部署在网关的安全设备.

以上研究和工作的前提是在互联网上划分出一个域，在这个域中按照事先设置的固定模式，集中管理或防御. 然而，互联网本身是一个开放、互通的网络空间，目前，以建立僵尸网络、发动分布式拒绝服务(Distributed Denial of Service，DDoS)攻击等为代表的群体性网络恶意行为大规模调用分布在整个互联网的资源[8]，他们位置分散，形式多样，仅对某个安全域的防护，无法深入分析这些群体性恶意行为. 另外，发生在不同地理位置、不同时间段的安全事件是否由同一组织所为，不同类别的安全事件是否存在着关联性，诸多安全事件信息结合在一起能否挖掘出更深层次的安全隐患.

针对上述问题，本文提出了一个基于通用图灵机的网络安全设备协同联动模型，并通过该模型实现了一个面向国家基础网络的网络安全协同联动系统. 为追踪、分析互联网的大规模安全事件提供了有力平台.

本文第2节介绍协同联动模型；第3节描述协同联动系统的实现；第4节阐述协同联动系统的应用实例和数据分析；第5节是应用实例；第6节总结全文.

2 基于通用图灵机的协同联动模型

2.1 通用图灵机和存储程序计算机

为了解决有限步数可计算问题，英国数学家阿兰·图灵于1937年提出了抽象计算模型——图灵机(Turing Machine)[9]. 图灵进一步描述了一种可存储指令的通用图灵机[10]，这种机器能够读取并执行不同的程序. 在某种意义上，通用图灵机相当于一个解释程序，给定任意应用程序及输入数据，利用这个解释程序可以得到输出. 采用通用图灵机模型可以对任意特定的图灵机进行仿真[11]. 图灵认为这样的一台机器能模拟人类进行任何计算过程，是可等价于任何有限逻辑数学过程的终极强大计算机器. 因此，通用图灵机是一个通用计算机的数学模型，其抽象意义为一种数学逻辑机，是一个普适性定义的理

想模型.约翰·冯·诺伊曼将这一思想作了具体实现,提出了存储程序计算机(stored program computer)的体系架构,把需要解决的问题用软件编制程序,然后将程序和数据都存放在存储器里,由中央处理器(CPU)根据指令对数据进行操作.这一思想被现代计算机的体系结构所继承.

2.2 协同联动模型形式定义

网络安全的协同联动工作是根据事先设定的工作任务,协调多种类型网络设备共同合作,利用这些设备提供的信息,挖掘、分析各种异常网络行为.如果把需要协同的设备看作外部设备,则对这些设备提供的信息进行协同处理过程可认为是进行有限步数的计算,因此,本文根据图灵的计算模型和存储程序计算机的体系结构提出了一个处置网络安全事件的协同联动模型(Coordinative Running Model, CRM).

定义 1. 协同联动模型的形式定义.一个协同联动模型是一个十元组:$C=(Q,q_0,q_{\text{accept}},q_{\text{reject}},\Sigma,\delta,\Gamma_1,\Gamma_2,\Gamma_3,\Gamma_4)$,其中:

Q 是状态集;

$q_0\in Q$ 是起始状态;

$q_{\text{accept}}\in Q$ 是接受状态;

$q_{\text{reject}}\in Q$ 是拒绝状态,且 $q_{\text{reject}}\neq q_{\text{accept}}$;

Σ 为输入字母表,不包括特殊空白符 B;

Γ_1 是事件带字母表,$B\in\Gamma_1$ 且 $\Sigma\subset\Gamma_1$;

Γ_2 是知识带字母表,$B\in\Gamma_2$ 且 $\Sigma\subset\Gamma_2$;

Γ_3 是结果带字母表,$B\in\Gamma_3$ 且 $\Sigma\subset\Gamma_3$;

Γ_4 是流程带字母表,$B\in\Gamma_4$ 且 $\Sigma\subset\Gamma_4$.

$\delta:Q\times\Gamma^4\xrightarrow{\delta}Q\times\Gamma^4\times\{L,R\}^4$ 是状态转移函数.把 $Q\times\Gamma^4$ 中的某一个元素映射为 $Q\times\Gamma^4\times\{L,R\}^4$ 中的一个元素.即定义域和值域分别为 $Q\times\Gamma^4$ 和 $Q\times\Gamma^4\times\{L,R\}^4$.

工作带 Γ_1 和 Γ_2 上分别记录需要处理的事件字母表和知识字母表;Γ_3 记录处理结果字母表;而 Γ_4 上记录着由不同的协同任务定制的工作流程字母表.

定理 1. 协同联动模型 CRM 等价于通用图灵机.

证明.

(1) CRM 等价于图灵机.定义 1 中,根据存储功能的不同,定义了多条工作带,显然,CRM 是多带图灵机,而多带图灵机与图灵机等价,因此,CRM 等价于图灵机.

(2) CRM 等价于通用图灵机.设 R_1,R_2,R_3,R_4 分别表示 C 每次计算时的事件字母表、知识字母表、结果字母表和流程字母表.图灵机 M 被定义为一个七元组,M 的格局由当前状态、当前带内容和读写头的位置构成.计算过程中,根据转移函数 δ 描述的规则,M 从一个格局转换到另一个格局,因此,图灵机的本质是一个算法或函数,给定一个输入数据 x,可计算出 $f(x)$.也就是说,M 相当于一个专用机,进行一种特定的计算.

而每一个协同任务就是要 CRM 根据编制的流程完成一次特定的计算.通过流程字母表 R_4 编制的流程确定一个算法 M_i.给定 R_1,R_2,根据算法 M_i 制定转移函数 δ 的规则,C 从一个格局转到另一个格局,因此,算法 M_i 实际就是一个图灵机.

也就是说,每个协同任务对应一个流程,而每个流程对应着一个图灵机.因此,图灵机 M_i 对输入 (R_1,R_2) 的计算等同于 CRM 对数组 (R_1,R_2,R_4) 的计算.即 CRM 输入任何图灵机 M_i 的编码,CRM 都可以仿真 M_i 的计算,CRM 相当于 M_i 的解释器,所以,CRM 等价于一个通用图灵机(可以完成任意的协同任务).

根据定理 1,我们构建的协同联动模型 CRM 是一个通用计算模型.为了能够物理实现 CRM,下面分别从使用者(包括一般的用户和系统开发人员)和系统结构的角度进一步阐述.

2.3 模型层次结构

从使用者角度,协同联动模型表现为层次结构,如图 1 划分为操作层、系统层和物理设备层.

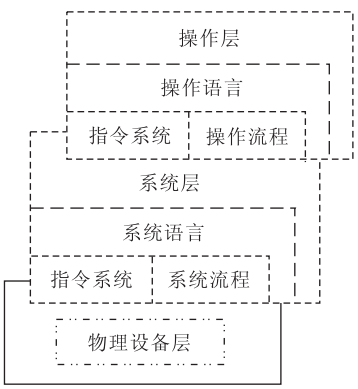

图 1 协同分析模型层次结构

(1)操作层,直接面向一般的操作人员,由以下相关概念来定义.

定义 2. 操作语言.是面向一般操作人员定义的一组语法规则.借助这组语法规则,使用者直接操

作协同联动系统，并准确完成相应的操作任务. 为便于操作人员使用，操作语言尽量采用简单指令或图形界面化的形式实现.

定义 3.　操作指令集. 在操作语言的语法框架下，定义的一组指示协同联动系统完成各种基本操作动作的指令. 多条操作指令按工作流程组合起来，以完成一定协同联动任务. 每条指令也可单独使用，执行指定的操作行为. 指令由操作码和变量码组成. 操作码，指示协同联动系统进行基本的操作动作；变量码，表示操作指令的执行对象以及操作对象的存储位置.

定义 4.　操作流程. 指操作人员根据要完成的协同联动任务，按照操作语言的语法规则，组合起来的一系列操作指令. 每一条指令指示了协同联动系统的每一步行为.

(2) 系统层主要面向协同联动系统的开发人员，由以下几个概念定义.

定义 5.　系统语言. 指协同联动系统的各组成设备可直接识别和解读的一组语法规则. 操作指令系统中的每条指令在系统语言的语法规则下都有若干条系统指令与之对应.

定义 6.　系统指令集. 在系统语言的语法框架下，最基本指令的集合.

定义 7.　系统流程. 在系统语言的语法框架下，按照协同联动模型要完成任务的逻辑顺序组合起来，系统可直接执行的系统指令集合.

(3) 物理设备层，是构建协同联动系统，执行协同联动任务的各种物理设备的总称.

在上述模型中，操作语言是源语言，系统语言是目标语言. 操作人员通过操作语言下达的指令和编写的操作流程要翻译为系统语言语法框架下的系统指令和系统流程，以驱动物理设备层的各组成设备. 也可以直接用系统语言编写系统流程完成协同分析任务.

2.4　模型系统结构

根据协同联动模型的形式定义，构建模型系统结构. 如图 2 所示，其结构由以下各系统部件组成.

(1) 接口系统(Interface System，IS). 是协同联动系统与外界信息交互的中转站. 需要协同的数据一般来自各种网络安全设备，也可能是合作组织或个人提供的网络事件信息，数据传输方式各不相同，而且数据格式和存储形式各异. 接口系统兼容多种通信方式，并统一不同来源的异构数据格式.

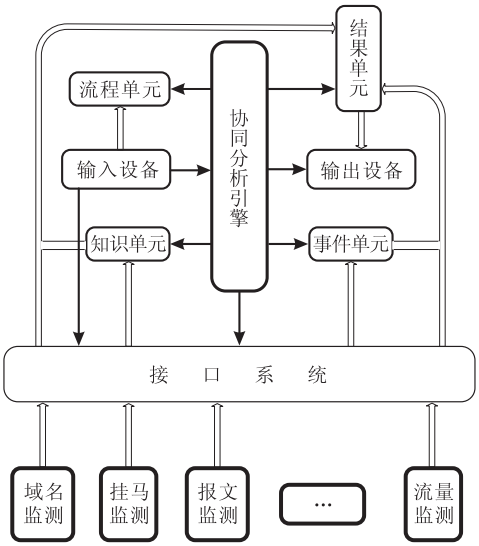

图 2　协同分析模型逻辑架构

(2) 存储部件(Storage Component). 根据协同模型形式定义中的不同工作带，存储部件按功能包括事件单元、知识单元、流程单元、结果单元等几部分. 每个存储单元的功能如下：

事件单元. 用于存储外界提供的网络安全事件信息. 外界数据通过接口系统将信息标准化后，送到事件单元中. 协同联动引擎可直接访问事件单元，按照工作流程操作事件单元中数据，并将结果存入结果单元中.

知识单元. 存储协同联动过程中所需的专家知识. 比如域名监测设备提供的 DNS 解析信息.

流程单元. 存储操作人员根据要完成的协同分析任务，通过操作语言或系统语言编写的工作流程. 每个流程设定编码.

结果单元. 存储协同联动任务的执行结果. 可随时通过输出设备展示结果单元中的数据.

(3) 协同联动引擎(Coordinative Running Engine，CRE)是协同联动模型的核心部件. 将操作人员下达的指令解释为协同联动系统可识别的系统语言，并按照流程或指令调度、协同各存储部件中的信息. 工作过程分为接受指令(receive)、解码翻译(decode)、执行流程(execute)、结果存入(write-back) 4 个阶段.

接受指令. CRE 可直接接受操作人员通过输入设备传入的操作指令，或调用流程单元中的操作流程.

解码翻译. CRE 接受操作指令后，首先按照操作语言的语法规则对每条指令解码，解析出操作码

和变量码;然后根据系统语言的语法框架把解码后的指令翻译成系统可识别的若干条系统指令;最后组合系统指令,编写系统流程进入执行流程阶段或设定系统流程编码存入流程单元.

执行流程. 解码翻译后的系统流程可直接执行,或者在操作人员的指令下按编号运行流程单元中的系统流程.

结果存入. 将各协同联动任务的执行结果写入结果单元中,按照工作流程编码,保存执行结果,以供在输出设备上展示.

(4)输入设备(Input Device). 它是操作人员与协同联动系统通信的桥梁,操作人员通过输入设备向系统输入指令和数据.

(5)输出设备(Output Device). 输出或展示操作人员的指令或流程的执行结果.

3 协同联动系统

基于协同联动模型,我们实现了一个开放、通用和易扩展的协同联动系统(Coordinative Running System,CRS),应用于基础网络的监测环境中.

3.1 存储部件

各种数据库或文件系统等都可以实现存储部件. 针对骨干网的海量信息,我们采用了具有强大数据处理能力的大型关系数据库系统作为主要存储部件,各存储单元根据不同的存储功能分别由不同的数据库实现. 为便于寻址和易于扩展,将数据表看作可操作的最小存储单位,定义为存储资源,并通过存储向量描述:

$$\langle MNum, Unit, DBType, Host, User, PWD, TNS, DBName, Table\rangle.$$

各分量含义分别为存储资源编号、存储部件类型、对应数据库的类型(如 ORACLE、DB2 等)、数据库主机 IP 地址、用户名、用户密码、ORACLE 连接选项、数据库名、数据表名. 其中 TNS 是 ORACLE 连接时使用的选项,对其它类型数据库填写 NULL 即可.

为了能快速寻址,高效维护存储资源,采用链表结构组织存储向量,构成存储地址表,通过 $MNum$ 快速连接数据库,定位到相应数据表.

3.2 接口系统

为了兼容多种通信协议和异构的数据格式,接口系统采用模块化设计结构,如图 3 所示.

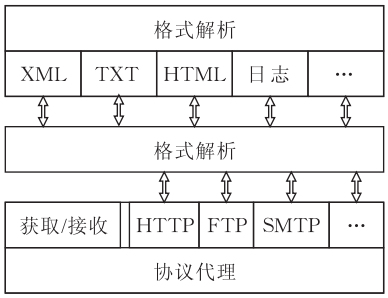

图 3 接口系统

CRS 从外部设备接收数据时,协议代理根据不同通信协议和连接方式(主动获取或被动接收)调用相应协议代理模块建立数据传输信道. 接收的数据临时存入数据区. 格式解析部分读取数据区中的数据,根据不同数据格式,应用不同格式解析模块,数据转换为 CRS 定义的标准数据格式. 系统向外部输出数据时与上述过程相反.

为了对外部设备统一管理,接口系统定义了外设资源向量:$\langle DNum, Protocol, Host, User, PWD, Format\rangle$. 各分量分别表示外设资源编号、外部设备的通信协议、外部设备 IP 地址、通信协议使用的用户名和密码以及外部设备使用的数据格式. 与某外设交互数据时,通过外设的唯一标识 $DNum$ 即可快速与其建立连接,通过向量中的 $Protocol$ 和 $Format$ 可便捷地匹配某外部设备需要的通信协议代理模块和数据格式解析模块. 所有的外设资源向量构成外设资源表.

3.3 协同联动引擎

协同联动引擎是系统的核心部件,解析操作指令,并翻译为系统可识别的系统语言. 为了便于一般使用人员的操作,我们设计了一个简单易用的脚本语言 CADL(Collaborative Analysis Description Language)作为系统的操作语言. 操作指令由基本指令和扩展指令两部分构成. 基本指令是协同联动最常用指令,分协同分析指令、数据交互指令和调度指令三类. 协同分析指令通过对存储部件的操作完成主要的协同分析工作;数据交互指令负责协调系统与外设通过接口系统的数据交互;调度指令协调工作流程的执行情况. 当有特定的或特别复杂的协同分析任务时,需要为其定制专门指令,也就是扩展指令. 表 1 中列出了典型的基本操作指令.

以操作指令为例,CADL 的文法用巴克斯范式(BNF)描述如下:

〈操作指令〉::=〈操作码〉:〈变量码〉;

〈操作码〉::=〈关键字〉；

〈变量码〉::=字符串|〈值〉；

〈关键字〉::=Table_Create|Table_Remove|Index_Creat|Index_Remove|Read_Write|Redun_Rem|……；

〈值〉::=字符串或数字.

由于存储部件的功能由数据库系统担任，并考虑到系统的执行效率，采用 SQL 和 C 语言作为系统语言. 一条操作指令的功能由一段 SQL 和 C 语言程序实现，并封装成函数. 在解码翻译阶段，解析操作指令，调用该指令对应的系统语言中的函数. 一段 CADL 编写的操作流程，经解码后，翻译为若干段 SQL 和 C 语言程序构成的系统流程.

表 1　典型操作指令

	指令	操作码	变量码	功能
协同分析指令	建存储资源	Table_Create	*MNum*：存储资源编号. *Unit*：存储部件名称. *Table*：存储表编号或名称. *Table_Define*：该存储表的描述信息.	在存储单元 *Unit* 中按照 *Table_Define* 建立新的存储资源 *Table*，并在存储地址表中添加新的存储向量.
	删存储资源	Table_Remove	*MNum*：存储资源编号.	删除指定的存储资源.
	建存储索引	Index_Creat	*MNum*：要建立索引存储资源的编号. *Name_Index*：索引名称. *Field_Index*：存储资源中的索引字段.	对指定的存储资源建立索引，以提高操作效率.
	删存储索引	Index_Remove	*MNum*：需要删除索引的存储资源编号. *Name_Index*：要删除索引的名称.	删除指定的索引.
	读写	Read_Write	*MNum*1，*MNum*2：要读取的存储资源的编号，可以是多个存储资源. *Field*1，*Field*2：要读取的存储资源中的字段，可以是多个字段. *MNum*3：要写入的存储资源的编号. *Field*31，*Field*32：要写入的存储资源中的字段，可以是多个字段.	读取 *MNum*1 中部分字段对应的数据写入到 *MNum*3 中；或读取 *MNum*1 与 *MNum*2 相同字段对应的数据，写入到 *MNum*3 中.
	数据去重	Redun_Rem	*MNum*1：需要去重的数据表对应的存储资源编号. *Field*1：*MNum*1 数据表去重依据的字段，可以是多个字段. *MNum*2：去重后的数据表对应的存储资源编号.	消除存储资源编号为 *MNum*1 数据表中的冗余数据，写入到存储资源编号 *MNum*2 的数据表中.
数据交互指令	输入	Input	*DNum*：外设资源编号. *MNum*：存储资源编号.	通过存储向量和外设资源向量将指定外部设备数据经过接口系统输入到指定存储部件中.
	输出	Output	*DNum*：外设资源编号. *MNum*：存储资源编号.	通过存储向量和外设资源向量将指定存储部件中的数据经过接口系统输出到指定外部设备.
调度指令	定时执行	Schedule	*Start*：开始时间. *End*：结束时间. *Interval*：时间间隔. *Procedure*：流程名称.	在开始时间 Start 和结束时间 End 的时间范围内，一个时间间隔 Interval，执行一次流程 Procedure.

4　相关工作比较与分析

多种类安全设备协同工作的典型系统有 UTM 和 SOC[7] 等. UTM 将防火墙、入侵检测、网关防病毒等集成在一起，是一种集成型网关设备. 与本文的协同联动模型在形式和实质上都完全不同.

SOC 协同管理多种安全设备，但目前业界对 SOC 还没有形成统一的理解，SOC 的实现方式也各不相同. 其一般模型的定位是采用划分安全域的思想，以资产为核心的一套资产风险模型. SOC 与本文提出的协同联动系统有相似的逻辑架构，但二者有本质区别.

SOC 是协助管理员进行事件分析、风险分析、预警管理和应急响应处理的集中安全管理系统. 其根本模型是 PDR 模型. 在一个安全域内针对不同时间、不同位置的离散单一安全事件进行收集、汇总、过滤和关联分析以形成统一的安全决策，对事件进行响应和处理，实现全域的资产风险管理.

本文提出的协同联动模型是一个开放式的、以图灵机模型为基础的抽象计算模型，在互联网空间追踪和分析大规模网络安全事件. 主要是协同不同种类的互联网设备，以这些设备提供的海量信息为数据源，分析危害网络空间的大规模网络安全事件. 表 2 列出了二者的比较.

表 2 协同联动模型与 SOC 比较

	协同联动模型	SOC 模型
面向对象	开放的互联网空间	安全域
基础模型	图灵机计算模型	闭环 PDR 模型
数据来源	分布在互联网的各种安全监测设备提供的海量信息	域中不同位置、分散的单一安全事件
主要目的	事件的分析、挖掘	域内资产风险管理
通用性	可根据不同的分析任务,编制不同流程	按照固定模板关联单一安全事件
操作方式	编制工作流程,需要学习语言	按模板配置安全策略,操作相对简单

SOC 按照一个大的规则库来管理安全域,是一套专家系统,存储了大量规则,但数量有限,有些规则没有被存储,这些规则涉及到的安全事件,SOC 无能为力.而基于图灵机的协同联动模型根据不同的协同任务定制不同的工作流程,应用计算概念,是一种用有限来应对无限的方法.

此外,协同联动系统完成一次协同联动任务实质是通用图灵机进行了一次计算,需要考虑图灵机进入停机状态的时间,即需要多长时间完成协同任务才能有实际应用价值.因此,协同联动系统在执行不同的协同任务时,要根据相应环境和不同的应用做效率上的优化.这就需要我们在设计操作语言和系统语言时考虑执行效率的优化功能.

5 应用实例

为了验证本文基于通用图灵机建立的协同联动模型 CRM 和协同联动系统 CRS 的自动协同联动能力,本节分析了 CRS 在国家基础网络的安全应用中 3 个典型实例,并给出每个实例基于通用图灵机的形式化描述.

5.1 僵尸网络的检测和追踪

这个协同任务要求协同 3 种不同类型的设备,是一个多元协同联动任务.

5.1.1 任务概述

僵尸网络[12]是目前互联网安全的一个严重威胁.采用 IRC 和 HTTP 协议通信的集中式僵尸网络[13]一般通过 DNS 实现控制服务器与受控主机间的联系,并经常通过更换控制服务器域名和 IP 的迁移机制[14]来躲避检测.因此,同一个僵尸网络的服务器域名和 IP 在不同时刻可能会不相同.另外,僵尸网络成员也在动态变化[15],一些受控主机可能摆脱僵尸网络的控制,或者有新的受控成员加入.为了更好地检测和追踪集中式僵尸网络的控制服务器和受控端主机,协同了以下几个设备共同分析僵尸网络行为.

(1)网络报文监测系统.国家网络安全监测平台(863-917 平台)[16],实时检测互联网中特定的安全事件,提供恶意代码的受控主机 IP、服务器 IP 和通信时间等信息.

(2)DNS 监测系统.记录国内各服务器的域名解析信息.

(3)流监测系统.提供指定 IP 与互联网其它主机通信的数据流信息.

协同过程如图 4 所示:

(1)报文监测系统根据僵尸程序特征码捕获活跃的僵尸程序样本,并发现控制端 IP 地址;

(2)DNS 监测系统通过监测到的 IP、域名对应关系,根据 IP 提供该僵尸网络控制端域名;

(3)DNS 监测系统进一步匹配一段时间内与控制端域名对应的所有 IP,给出控制端 IP 列表;

(4)将控制端 IP 列表与该时间段内流监测系统日志关联,获得与僵尸网络控制端通信的受控僵尸主机 IP 地址列表.

定时重复执行上述过程,可以获得更多的控制服务器 IP 和受控主机 IP,追踪、分析集中式僵尸网络在不同时间的变化状况.

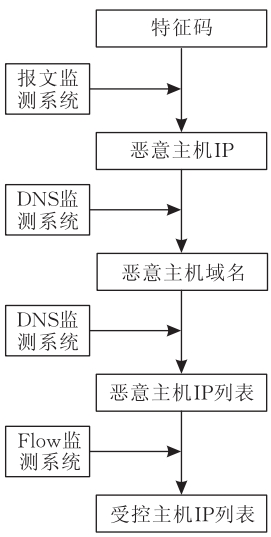

图 4 僵尸网络协同分析

5.1.2 流程设计

根据这个协同任务,规划操作流程如表 3 所示.通过 c,d 消除原始数据中冗余信息;通过 e,f 建立存储索引,提高对存储资源的搜索效率.操作人员按照表 3 中各步骤,通过 CADL 语法编写操作流程,存入流程单元,协同联动引擎将其翻译为系统流程并执行.操作人员也可单步执行操作流程中的语句,逐步完成协同分析任务.

表 3　僵尸网络协同分析操作流程

步骤	操作码	变量码	执行内容
a	Input	*DNum*1,*MNum*1	将报文监测系统提供的僵尸网络控制端 IP 信息写入事件单元的存储资源 *MNum*1 中.
b	Input	*DNum*2,*MNum*2	将 DNS 监控系统提供的域名、IP 对应信息写入知识单元的存储资源 *MNum*2 中.
c	Redun_Rem	*MNum*1,*Field*1,*MNum*3	根据字段 *Field*1 消除存储资源 *MNum*1 中的冗余记录,去重后的结果写入到事件单元的存储资源 *MNum*3 中.
d	Redun_Rem	*MNum*2,*Field*1,*MNum*4	根据字段 *Field*2 消除存储资源 *MNum*2 中的冗余记录,去重后的结果写入到知识单元的存储资源 *MNum*4 中.
e	Index_Creat	*MNum*3,*Name_Index*1,*Field_Index*1	对事件单元中 *MNum*3 按照 *Field_Index*1 建存储索引 *Name_Index*1.
f	Index_Creat	*MNum*4,*Name_Index*2,*Field_Index*2	对知识单元中 *MNum*4 按照 *Field_Index*2 建存储索引 *Name_Index*2.
g	Read_Write	*MNum*3,*MNum*4,*Field*3,*Field*4,*MNum*5	读取存储资源 *MNum*3 的 *Field*3 字段(僵尸网络控制端 IP 字段)和存储资源 *MNum*4 中 *Field*4 字段(IP 字段)相同的记录,写入事件单元的 *MNum*5 中(获取僵尸网络控制端 IP 对应的域名).
h	Read_Write	*MNum*5,*MNum*4,*Field*5,*Field*6,*MNum*6	读取事件单元的 *MNum*5 中的 *Field*5 字段(僵尸网络控制端域名字段)和知识单元的 *MNum*4 中 *Field*6 字段(域名字段)相同的记录,写入事件单元的 *MNum*6 中(获取控制端域名对应的 IP 列表).
i	Read_Write	*MNum*6,*MNum*7,*Field*7,*Field*8,*Field*9	读取 *MNum*6 中的 *Field*7(域名字段)相同的记录,并获取 *Field*8 字段(IP 字段)和 *Field*9(时间字段)值,写入到事件单元的 *MNum*7 中(记录下同一个控制端域名对应的 IP 列表和时间段).
j	Output	MNum7,DNum3	把 *MNum*7(有时间段的 IP 列表)传给设备 *DNum*3(数据流监测系统).
k	Input	*DNum*3,*MNum*8	把设备 *DNum*3(数据流监测设备)传过来的数据(控制端 IP 列表和受控端 IP 列表)写入到结果单元的 *MNum*8 中.
l	Index_Creat	*MNum*8,*Name_Index*3,*Field_Index*3	对 *MNum*8 按照 *Field_Index*3 建立存储索引 *Name_Index*3.

最后,通过 Schedule 指令每隔 6h 重复执行这个过程,进一步追踪由于僵尸网络迁移和动态变化产生的新僵尸网络控制端和僵尸网络成员.

5.1.3　形式化描述

首先将这个任务的协同过程符号化. a,c,e 和 b,d,f 分别是报文监测设备和 DNS 监控系统提供数据的写入、去重和寻址优化 3 个步骤,顺序不能改变. g,h,i,j,k,l 几个步骤也须固定不变. 因此这个协同过程可以译码为符号串 $N=(\mathrm{K\ g\ h\ i\ j\ k\ l})$,其中 K 表示分别固定 a,c,e 和 b,d,f 两组字符的顺序后的 6 个字符的任意组合构成的字符串集合. 因此 N 表示了这些字符组合后的所有字符串. 符号串中的符号取自有穷的字母表 Σ,这些字符串就构成了语言 L. 所以,问题转化为构造一个识别语言 L 的图灵机 M_1.

最后给出这个实例中图灵机 M 的形式描述. $M=(Q,\Sigma,\Gamma,q_0,q_{\mathrm{accept}},q_{\mathrm{reject}},\delta)$,其中:

$Q=\{q_1,q_2,\cdots,q_{11},q_{\mathrm{accept}},q_{\mathrm{reject}}\}$;

$\Sigma=\{\mathrm{a,b,c,d,e,f,g,h,i,j,k,l}\}$;

$\Gamma=\{\mathrm{a,b,c,d,e,f,g,h,i,j,k,l,B}\}$;

$\delta: Q\times\Gamma\xrightarrow{\delta}Q\times\Gamma\times\{\mathrm{L,R}\}$;

q_0 是起始状态;

q_{accept} 是接受状态;

q_{reject} 是拒绝状态.

开始状态为 q_0,把要判断的符号串写在工作带上,每格写入一个符号,其余空格是空白符号 B. 对于输入符号串 $N=(\mathrm{K\ g\ h\ i\ j\ k\ l})$,从左向右扫描,每扫描一个符号说明经过了一次协同引擎的处理,解析出了新的信息,生成一个新的状态 q_i.

状态图如图 5 所示. 从 q_0 到 q_1 的转移记为 a→c,R,即转移函数 $\delta(q_0,\mathrm{a})=(q_1,\mathrm{c,R})$,表示当状态处于 q_0 且读写头读符号 a 时,机器的状态变为 q_1,写下符号 c,并向右移动读写头. 其它状态转移与此类似. 为了图示清晰,图中没有标出每个状态到拒绝状态的转移标记,而且单独列出了需要交叉连线的状态转移标记.

5.1.4　执行结果分析

(1) CRS,一个时间段执行结果. 表 4 列出了某一时间段内,各设备提供的数据量及协同联动后的结果. CRS 通过去重指令有效消除了冗余信息,DNS 监测系统的数据尤为明显,消除了近 30%. 共分析出 10 万多条恶意主机 IP,受控端主机有近 6 千万条记录,有效地分析出僵尸网络的服务器和受控主机的相关信息. 为进一步验证恶意主机信息我们进行了(2)、(3)两步观察分析.

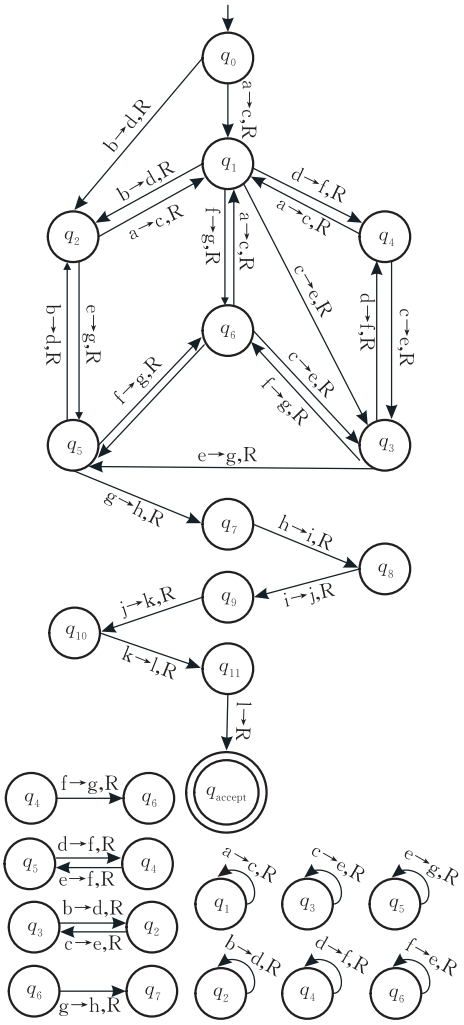

图 5 僵尸网络协同分析状态图

表 4 一个时间段内数据量

数据来源	数量	
	去重前	去重后
863-917 平台	58619	50486
DNS 监测系统	18298304	11198870
恶意主机域名	35536	
恶意主机 IP	109578	
流监控返回信息	58346396	

(2)某僵尸网络一个时间段协同结果. 在一个时间段内,一个典型僵尸网络经过协同联动的变化情况如表 5 所示. CRS 根据控制端 IP *.*.*.46 发现了该僵尸网络的一个域名①和另一个控制服务器 IP,这两个 IP 都与 12 万个以上的受控客户端有数据通信行为.

表 5 某僵尸网络观察情况

控制端 IP	域名	控制端 IP 列表	受控端 IP 数量(条)
..*.46	jj.*.*.su	*.*.*.71	127798
		..*.76	122474

(3)某僵尸网络多时间段协同结果. 从表 6 中看出,在 72h 后,CRS 协同出新的控制端 IP,受控端主机的数量也随之明显增加;120h 后,服务器的域名产生变化,僵尸网络发生了迁移行为,并且受控主机的数量在增加;在 240h 后,再次发现新迁移行为,僵尸网络规模继续扩大. 这是个活跃的僵尸网络,在其发展过程中,应用了多个服务器 IP,把受控主机分成多个群体,并发生了 2 次迁移. 一个正常服务器不会频繁发生更换域名的迁移行为,因此,多个时间段的观察结果进一步确认了僵尸网络检测信息.

表 6 几个时间段的观察情况

时间	控制端 IP	控制端域名	受控端 IP 数量(条)
初始	*.*.*.78	oi.*.*.net	1364
24h 后	*.*.*.78	oi.*.*.net	1289
72h 后	*.*.*.78 *.*.*.59 *.*.*.96 *.*.*.62	oi.*.*.net	6986
120h 后	*.*.*.78 *.*.*.59 *.*.*.157 *.*.*.62	gred.*.*.net dbs.*.com	12364
240h 后	*.*.*.166 *.*.*.59 *.*.*.157 *.*.*.62	eyeblaster.*.*.com tjqvod.*.*.com	149686

(4)效率比较. CRS 每次执行这个操作流程大约需要 51min,我们同时比较了没有去重指令和建索引指令的操作流程运行时间,用时 194min. 因此,如第 4 节所述,只有通过效率优化,考虑进入停机状态的时间,基于图灵机构建的模型才有实际应用价值.

通过以上 4 个方面的分析,可以看出,CRS 能有效完成追踪大规模僵尸网络的协同联动任务. 消除海量数据的冗余信息,挖掘僵尸网络在不同时间段的变化情况,同时,也为确认僵尸网络的检测结果提供了有效分析手段. 并且,通过高效的实施协同联动任务,提高了安全事件的处置效率.

5.2 大规模 DDoS 攻击事件关联

这是一个一元协同任务,要求对同一类型的几个网络安全监测设备做协同分析.

5.2.1 任务概述

分布式拒绝服务攻击具有地理分布性和一定时间延续性,可利用分布式实时监测设备检测 DDoS 攻击行为. 但对大规模 DDoS 行为,设备上报的信息量巨大,需要关联这些信息,掌握 DDoS 攻击的真实情况.

利用 CRS,协同分析了一套分布式网络恶意事件

① 通过 Google 搜索确认,该域名为一个控制服务器位于俄罗斯的僵尸网络域名.

监测系统上报的 DDoS 攻击. 这套系统部署在国内几个基础网络节点和国际出入口，实时上报蠕虫、僵尸网络、木马和 DDoS 等各类恶意行为. 协同步骤如下：

(1) 提取一个时间段(比如 24h)的各类安全事件原始数据；

(2) 选出该时间段内的 DDoS 攻击事件信息；

(3) 时间关联. 合并同一攻击的连续报警信息，并统计攻击的数据量；

(4) 空间关联. 合并不同地点的监测设备在一个时间间隔内检测到同一 DDoS 攻击信息，并统计攻击的总数据量，计算攻击强度和攻击规模.

5.2.2　流程设计

与 5.1.2 节过程相似，限于篇幅，本文不再详述. 工作流程如表 7 所示.

表 7　DDoS 去重分析操作流程

步骤	操作码	变量码	执行内容
a	写入 Input	*DNum*1…*DNumn*, *MNum*9	导入各安全事件监测设备 *DNum*1…*DNumn* 的数据到 CAS 系统事件单元的 *MNum*9.
b	Read_Write	*MNum*9, *MNum*10, *Field*1(时间字段)	从 *MNum*9 中读取按照 *Field*1 字段标识的一段时间内的安全事件信息，写入到事件单元的 *MNum*10.
c	Read_Write	*MNum*10, *MNum*11, *Field*2(事件标识字段)	读取 *MNum*10 中按 *Field*2 字段标识的 DDoS 事件信息，写入到 *MNum*11.
d	Redun_Rem_time	*MNum*11, *MNum*12, *Field*3, *Field*2	在 *MNum*11 中，按设备标识字段 *Field*3 和时间字段 *Field*2 对各设备合并一个时间间隔内的相同信息，写入到 *MNum*12.
e	Redun_Rem_space	*MNum*12, *MNum*13, *Field*3, *Field*2	按设备标识字段 *Field*3 和时间字段 *Field*2 合并 *MNum*12 中相近时间的各设备的相同信息，写入到 *MNum*13.
f	Read_Write	*MNum*13, *MNum*14	将 *MNum*13 中执行结果写入到结果单元 *MNum*14.

5.2.3　形式化定义

先符号化这个协同过程：$N=(\mathrm{a\ b\ c\ d\ e\ f})$，符号串中的各字符顺序固定. 符号取自有穷的字母表 Σ，这个字符串构成了语言 L. 因此，构造一个识别语言 L 的图灵机 M_2. 开始状态为 q_0，状态图如图 6 所示(省略所有到拒绝状态的转移标记).

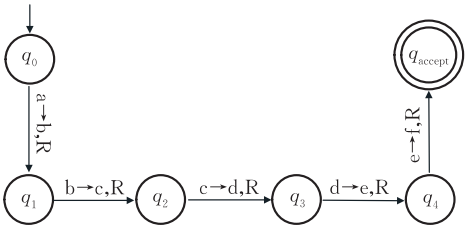

图 6　DDoS 事件关联分析状态图

最后给出这个实例图灵机 M 的形式描述.

$M=(Q,\Sigma,\Gamma,q_0,q_{accept},q_{reject},\delta)$，其中：

$Q=\{q_1,q_2,q_3,q_4,q_{accept},q_{reject}\}$；

$\Sigma=\{\mathrm{a,b,c,d,e,f}\}$；

$\Gamma=\{\mathrm{a,b,c,d,e,f,B}\}$；

$\delta: Q\times\Gamma\xrightarrow{\delta}Q\times\Gamma\times\{\mathrm{L,R}\}$；

q_0 是起始状态；

q_{accept} 是接受状态；

q_{reject} 是拒绝状态.

5.2.4　执行结果分析

为了说明 CRS 执行这个协同任务的意义，我们分析了 2010-03-03 到 2010-03-10 8 天的运行结果.

图 7 展示了关联前后的 DDoS 攻击事件数量的变化情况. 由于把同一事件的信息进行了合并，关联后明显减少了重复上报的 DDoS 事件冗余信息. 图 8 展示了关联前后，每天发生的 DDoS 攻击事件

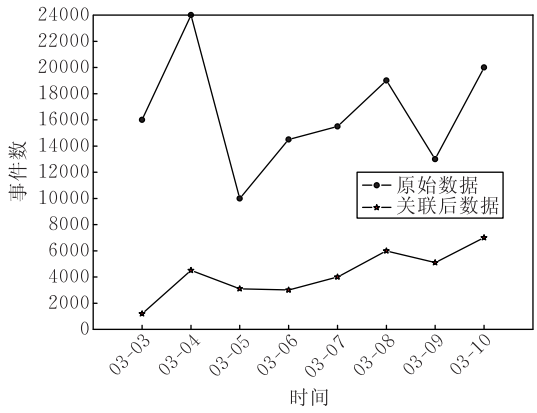

图 7　DDoS 事件关联结果图

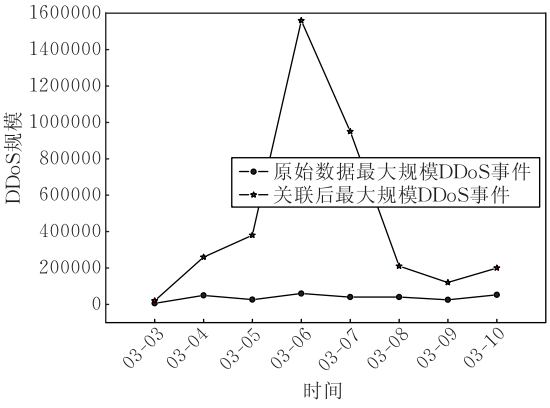

图 8　每天发现最大规模 DDoS 事件示意图

中最大规模的主机数量.经过关联,归并了参与同一次 DDoS 攻击事件的主机,主机数量显著增加,因此,更明确了一次 DDoS 攻击的规模和强度.

从以上结果可以看出,CRS 执行的这个协同任务能有效地从不同空间和时间的海量安全事件信息中关联出同一 DDoS 攻击事件,并分析出攻击规模和攻击强度,有效加强了大规模 DDoS 攻击的检测能力,为掌握 DDoS 攻击的真实情况提供了有力的支持.

5.3 僵尸网络与 DDoS 攻击源关系分析

这个任务是把 5.1 节追踪到的僵尸网络与 5.2 节关联出的 DDoS 攻击源主机群做进一步关联分析,是一个二元协同任务.

5.3.1 任务概述

DDoS 的攻击源是互联网空间内的大批受控傀儡主机,它们可能隶属于某一僵尸网络.因此,有必要分析 DDoS 攻击源的主机群与一个已知僵尸网络之间的关系.最直接的分析方法是计算两批主机的 IP 重合度.但由于很多互联网用户使用动态 IP,直接计算有很大误差,而 ISP 分配的动态 IP 具有一定的局部性,因此,我们用 C 类 IP 段代替单独的 IP 地址来计算 IP 重合度.

这个协同任务按如下步骤进行:

(1)分别读取 5.2 节 DDoS 攻击源主机列表和 5.1 节受控僵尸主机列表;

(2)分别统计两个列表中 C 类 IP 段的数量;

(3)统计两个列表中相同 IP 段的数量.

对任意两个要比较的主机列表,可通过调度指令,重复运行上述工作流程进行关联.

5.3.2 流程设计

与 5.1.2 节过程相似,限于篇幅,本文不再详述.工作流程如表 8 所示.

表 8 DDoS 僵尸网络协同分析操作流程

步骤	操作码	变量码	执行内容
a	Read_Write	*MNum*14, *MNum*15	读结果单元中 DDoS 事件表 *MNum*14,写入知识单元 *MNum*15.
b	Read_Write	*MNum*8, *MNum*16	读僵尸网络受控主机表 *MNum*8,写入知识单元 *MNum*16.
c	Count_IP	*MNum*15, *MNum*17 *MNum*16, *MNum*18 *Field*1(IP 字段)	合并 *MNum*15 中同一个 C 类 IP 段的 IP,写入 *MNum*17,合并 *MNum*16 中同一个 C 类 IP 段的 IP,写入 *MNum*18.
d	Count	*MNum*5, *MNum*6, *MNum*7, *Field*2(IP 段)	分别统计 *MNum*17 和 *MNum*18 中 IP 段数量以及相同 IP 段数量,结果写入 *MNum*19.
e	Read_Write	*MNum*7, *MNum*8	读取 *MNum*19,写入结果单元 *MNum*20.

5.3.3 形式化定义

将这个任务的协同过程符号化.a,b 两个步骤没有先后顺序.其余步骤固定不变.因此 这个协同过程可以译码为符号串 $N=(\mathrm{K\ c\ d\ e})$,其中 K 表示 a,b 或 b,a.因此 N 表示这些字符组合后的所有字符串.符号串中的符号取自有穷的字母表 Σ,这些字符串构成语言 L.因此,问题转化为构造一个识别语言 L 的图灵机 M_3.状态图如图 9 所示.图中省略了所有到拒绝状态的转移标记.

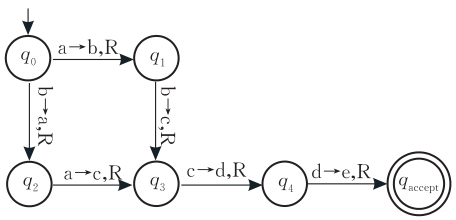

图 9 DDoS 与僵尸网络协同分析状态图

这个实例的图灵机 M_3 的形式描述为

$M=(Q,\Sigma,\Gamma,q_0,q_{\mathrm{accept}},q_{\mathrm{reject}},\delta)$,其中:

$Q=\{q_1,q_2,q_3,q_4,q_{\mathrm{accept}},q_{\mathrm{reject}}\}$;

$\Sigma=\{\mathrm{a,b,c,d,e}\}$;

$\Gamma=\{\mathrm{a,b,c,d,e,B}\}$;

$\delta: Q\times\Gamma\xrightarrow{\delta}Q\times\Gamma\times\{\mathrm{L,R}\}$;

q_0 是起始状态;

q_{accept} 是接受状态;

q_{reject} 是拒绝状态.

5.3.4 执行结果分析

为了说明执行这个协同任务的意义,我们分析了以下 3 种典型的情况.

(1)僵尸网络和 DDoS 攻击源部分重合.表 9 列出了两个实例.重合度越高,说明 DDoS 攻击源来自这个僵尸网络的可能性越大.而 DDoS 攻击源中没有与僵尸网络重合的主机,可能是属于这个僵尸网络而没有被检测到的受控主机.因此,这个协同任务加强了我们追踪僵尸网络的能力,有助于我们更深入了解僵尸网络规模.

表 9 DDoS 攻击源与僵尸网络重合情况

	僵尸主机数量	DDoS 攻击源数量	相同数量	占 DDoS 攻击源比/%	占僵尸网络比/%
1	122355	2196	656	29.87	0.05
2	93756	2818	809	28.7	0.09

(2)几次 DDoS 攻击事件的攻击源都与同一个僵尸网络有重合.表 10 统计了一个 134465 台受控主机的僵尸网络与 3 批 DDoS 攻击源主机的重合情况.每个 DDoS 攻击源都有部分主机与僵尸网络受控主机群重合,因此,这 3 批 DDoS 攻击源可能都来自这个僵尸网络,扩大了该僵尸网络规模.

表 10　一个僵尸网络与 3 个 DDoS 攻击源重合

	DDoS 攻击源主机数量	相同主机数量	占 DDoS 攻击源比率/%
1	1156	342	29.58
2	1096	305	27.8
3	1655	822	49.7

(3)在同一次 DDoS 攻击中发现攻击源来自不同的僵尸网络.表 11 是一个有 2696 台主机的 DDoS 攻击源与 3 个僵尸网络的重合情况.由于每批僵尸主机群都与同一 DDoS 攻击源有部分重合,说明它们可能是同一僵尸网络的不同部分,有助于我们分析僵尸主机群之间的关系.

表 11　一个 DDoS 攻击源与 3 个僵尸网络重合

	僵尸主机数量	相同主机数量	占僵尸主机比率/%
1	105636	445	0.042
2	96452	467	0.048
3	24556	946	0.385

从以上 3 个观察结果可以看出,CRS 运行的这个协同任务,为我们掌握僵尸网络的规模,追踪和分析僵尸网络之间的关系都提供了有效手段.因此,CRS 可为分析不同时间、不同空间安全事件间的关系,挖掘各事件关联后的更深层次安全隐患,提供有力的工作平台.

6　结　论

本文在讨论分析大规模网络安全事件的目的和意义的基础上,提出了一个基于通用图灵机的协同联动模型,并实现了协同联动系统,给出了相关的定义,系统在国家基础网的监测环境中进行了应用.与以前工作相比,这个模型在开放的网络空间中提供了网络安全协同联动的有力平台:(1)能够有效地追踪大规模网络安全事件在不同时间段的变化情况.通过高效地实施协同联动工作,提高了安全事件的处理效率.(2)从不同空间和不同时间的安全事件信息中关联出同一事件,并分析出事件的规模和强度,掌握事件的真实情况.(3)可以自动挖掘出不同安全事件间的关系,有助于更深入分析安全事件的本质.

致　谢　真诚感谢审稿人为本文提出的宝贵意见!真诚感谢 CNERT/CC 相关工作人员对我们工作的支持!

参　考　文　献

[1] Snapp S R, Brentano J, Dias G V, Goan T L, Heberlein L T, Ho C L, Levitt K N, Mukherjee B, Smaha S E, Grance T, Teal D M, Mansur D. DIDS(distributed intrusion detection system)—Motivation, architecture, and an early prototype//Proceedings of the 14th National Computer Security Conference. Washington D. C., 1991: 167-176

[2] White G B, Fisch E A, Pooch U W. Cooperating security managers: A peer-based intrusion detection system. IEEE Network, 1996, 10(1): 20-23

[3] Porras P A, Neumann P G. EMERALD: Event monitoring enabling responses to anomalous live disturbances//Proceedings of the 12th National Computer Information Systems Security Conference. Baltimore, Maryland, USA, 1997: 353-365

[4] Asaka M, Taguchi A, Goto S. The implementation of IDA: An intrusion detection agent system//Proceedings of the 11th Annual FIRST Conference 1999. Brisbane, AU, 1999: 146-160

[5] Cuppens F. Cooperative intrusion detection//International Symposium on Information Superiority: Tools for Crisis and Conflict-Management. Paris, France, 2001: 262-274

[6] Cuppens F. Managing alerts in a multi-intrusion detection environment//Proceedings of the 17th Annual Computer Security Applications Conference. New Orleans, USA, 2001: 22

[7] Renaud Bidou. Security operation center concepts & implementation. http://www.iv2-technologies.com/~rbidou/SOC Concept And Implementation.pdf, August 1, 2005

[8] Dagon D, Zou C, Lee W. Modeling botnet propagation using time zones//Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS 2006). San Diego, CA, 2006: 235-249

[9] Turing A M. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 1936, 42(2): 230-265

[10] Sipser M. Introduction to the Theory of Computation. 2nd Edition. Boston MA USA: Course Technology, 1997

[11] Zhang Li-Ang. Introduction to Computability and Complexity of Computational. 2nd Edition. Beijing: Peking University Press, 2004(in Chinese)
(张立昂. 可计算性与计算复杂性导引. 第 2 版. 北京:北京大学出版社, 1996)

[12] Rajab M A, Zarfoss J, Monrose F, Terzis A. A multifaceted approach to understanding the botnet phenomenon//Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC 2006). Rio de Janeiro, Brazil, 2006: 41-52

[13] Zhuge Jian-Wei, Han Xin-Hui, Zhou Yong-Lin, Ye Zhi-Yuan, Zou Wei. Research and development of botnets. Journal of Software, 2008, 19(3): 702-715(in Chinese)
(诸葛建伟，韩心慧，周勇林，叶志远，邹维. 僵尸网络研究. 软件学报，2008，19(3)：702-715)

[14] Zou C, Cunningham R. Honeypot-aware advanced botnet construction and maintenance//Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN'06). Philadelphia, PA, 2006: 199-208

[15] Rajab M A, Zarfoss J, Monrose F, Terzis A. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging//Proceedings of the 1st Conference on First Workshop on Hot Topics in Understanding Botnets. Cambridge, MA, USA, 2007: 5-5

[16] National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), National Certificate of Network and Information Technology-Management Center of China (NTC-MC). Emergency and Practice Guideline of Network Security. Beijing: Publishing House of Electronics Industry, 2008(in Chinese)
(国家计算机网络应急技术处理协调中心(CNCERT/CC)，全国网络与信息技术培训项目管理中心(NTC-MC). 网络安全应急实践指南. 北京：电子工业出版社，2008)

ZANG Tian-Ning, born in 1978, Ph. D. candidate. His research interests include botnet and coordinative analysis.

YUN Xiao-Chun, born in 1971, Ph. D., professor, Ph. D. supervisor. His research interests include network security, Internet model and Internet measurement.

ZHANG Yong-Zheng, born in 1978, Ph. D., associate professor. His research interests include network security and network security evaluation.

MEN Chao-Guang, born in 1963, Ph. D., professor, Ph. D. supervisor. His research interests include dependability computing etc.

SUN Jian-Liang, born in 1984, M. S. candidate. His research interests focus on coordinative analysis.

Background

With the fast development of information technology, various network malicious behaviors are increasingly emerging, such as DDoS, set up Botnet, etc. Critical IT systems are still highly vulnerable to these computer-based attacks. The attacks over the Internet may cause very serious damage at any time. Therefore, the Internet security is regarded as the most important thing, and a novel and effective solution is required to handle the increasingly coming malicious behaviors.

Some work proposed solutions by associate many network devices, such as alert aggregation, alert correlation, and so on. However, these work focused on the distributed intrusion detection system. The concept of the Security Operating Center (SOC) is introduced for the improvement of network management and security management. The United Threat Management (UTM) is also presented for protecting the enterprise network. Most of these investigations were based on the specific close network domain. The Internet consists of huge amount of resources including computing, storage and communications systems. It is serving as the underlying infrastructure for various areas. The aim of our work is to solve some critical problems about modeling and analyzing large-scale security events in the Internet.

In this paper, a model based on Universal Turing Machine for cooperatively analyzing the attacking events is proposed. This model coordinate the detecting and monitoring devices distributed in the back-bone network, and makes them work together to cooperatively deal with the threatening behaviors. The large-scale network security event can be tracked and analyzed by using this method. Base on this model, a collaborative running system is implemented for the safety of the back-bone network. The analysis results of typical security incidents data show that this system is efficient and effective to cooperatively analyze the large-scale security events. Currently, the system is running steadily in the monitoring environment of the back-bone network.

This paper is supported by the National High Technology Research and Development Program (863 Program) of China under grants of 2007AA01Z444, 2007AA01Z467, 2007AA01Z474 and 2007AA010501, and the National Natural Science Foundation of China under grants of 60703021 and 60873138. The goal of these projects is to improve the security of the backbone network.