基于知识图谱的网络安全威胁预警系统

1. 简介
   1. 背景分析

纵观整个2017年，网络安全防线多次面临勒索病毒事件的重大冲击。2017年5月WannaCry勒索病毒爆发，至少150个国家、30万名用户中招，造成损失达80亿美元。此后，勒索病毒持续活跃，6月份Petya勒索病毒席卷欧洲多个国家，政府机构、银行、企业等均遭大规模攻击。10月份“Bad Rabbit”（坏兔子）勒索软件导致“东欧陷落”，包括乌克兰与俄罗斯在内的东欧公司受灾严重。

根据腾讯公司提供的数据，2017全年总计已发现敲诈勒索病毒样本数量在660万个，平均每月检测到敲诈勒索病毒数量近55万个。Q3季度为4个季度中检测病毒的高峰，检测量为180万个，严重威胁到普通用户和企业用户的上网安全。

如果能够在病毒爆发初期，根据勒索病毒扫描445端口、关闭系统特定进程等典型行为，学习出勒索病毒的模式来预知安全威胁并进行威胁预警，避免其他用户感染病毒，那么就不会发生如此大范围的病毒感染，给人们带来巨大的损失。

另一方面，Gartner（国外跟踪前沿技术的网站：https://www.gartner.com）提出：

With the number connected devices predicted to pass 20 billion in the next three years, it only makes sense to leverage them as links to physical processes and products, allowing enterprises to easily monitor current status, collect valuable data and deploy perventive maintenance tools.

预计在未来三年内联网的设备将超过200亿，允许企业轻松监控设备状态、收集有价值的数据并部署预防性维护工具将成为一大技术热点。如时下正火的云计算、云存储，都要求监控庞大的计算机集群。

对计算机集群的监控，在安全方面实现在主机遭受攻击后识别攻击、预测攻

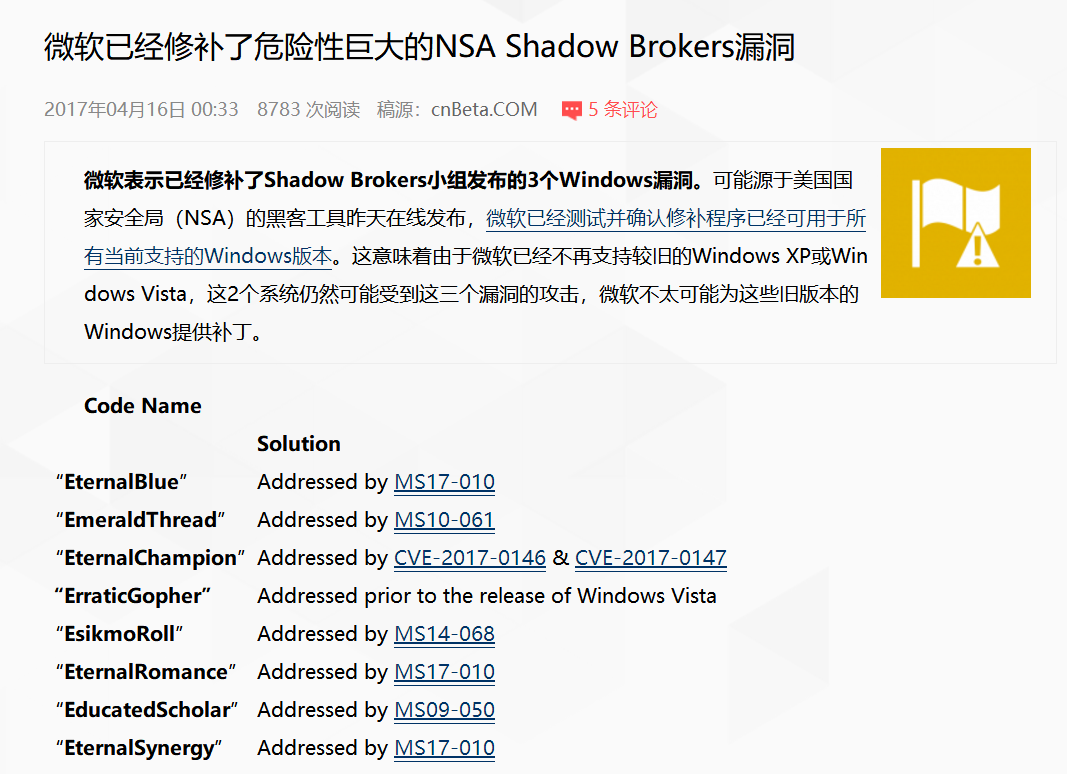
击行为并进行网络隔离防止病毒传播，在性能方面能及时处理大量数据是我们的系统的主要工作。

* 1. **相关工作**

目前的病毒防范尤其是新病毒的防范措施主要由操作系统（如微软）公司修补新漏洞，也有极少数极客在网上分享自制杀毒软件或相应简单措施的经验教程，以及相关应对网络安全威胁措施的研究报告。我们在分析了以下解决方案的优劣之后提出了我们的解决方案——基于知识图谱的网络安全威胁预警系统。

1. **国内外现有的解决方案**

（1）操作系统公司修补新漏洞——提供系统保障。让操作系统修复自身已被发现的漏洞。致力于打造一个以漏洞更少、安全性更高、及时更新漏洞修复补丁、为每一台使用该系统的主机提供系统安全的正版操作系统。如下图微软修补了危险性巨大的NSA Shadow Brokers漏洞。



（2）极客经验分享——提供自定义服务功能。极客在各种开源社区分享其网络安全技术知识和实战经验，通过教程教授网民小型自制杀毒软件或补丁，让网民自己定义功能服务，并开发出最适合自身需要的小型杀毒软件。当然对于大部分普通网民来说更现实易懂的方式就是上百度经验查一查教程，比如如下针对勒索病毒的简单端口关闭防范教程。



1. 应对网络安全威胁措施的研究报告

如上海交通大学的一篇硕士专业学位论文《基于数据挖掘的网络安全态势分析》，其中提到了使用数据挖掘技术通过分析大量数据进行模型分析和攻击检测。其中指出了利用关系型数据库记录并分析大量数据进行分析处理。

1. 现有解决方案的特点

优点：

（1）功能获取免费且简单，较快获得安全服务。微软等公司也会更新漏洞补丁发布市场，这些都是免费的，一键安装即可，大大的节省了人力物力，减少了流程，为用户减少了一定的损失。

1. 漏洞检测服务，可自定义检查周期。微软公司往往会提供漏洞检测、病毒扫描服务，帮助用户找到电脑中那些存在的且和现有最新病毒库或漏洞库相同的漏洞病毒，便于用户及时查杀或修补漏洞。
2. 维护成本低，通过微软等公司在不断更新病毒库和软件版本，微软公司不断发布新的漏洞并推出系统补丁，同时还有极客的技术分享，从操作系统公司、开源社区提供技术支持，以便用户保护其操作系统。

缺点：

1. 更新时间较慢。存在部分漏洞存在数年才解决的情况，危险性极高。一旦被黑客发现并加以利用则会造成巨大损失。
2. 过多的漏洞补丁会拖慢系统速度。就连微软公司自己也承认了这一点。过多的补丁会让电脑运行不堪重负，速度明显下降。
3. 关系型数据库的局限性。关系型数据库在面对复杂数据关系的时候应对显得吃力，即表与表之间的复杂关系出现的时候尤为明显。
4. 对未知病毒和存在的未知漏洞的防御能力低下。新的病毒总是在攻击之后才能开始思考对策进行漏洞修复，事后处理无法降低损失，更无法避免损失。
5. 对可能攻击方式缺乏逻辑上的梳理和扩展能力。承接低下的防御能力，现有机制无法将不同攻击方式，不同病毒之间的关联关系联系起来，即无法触类旁通，举一反三。倘若使用知识图谱一类的工具，则可以串起整个攻击流程图。
   1. **特色描述**

解决上述产品或者针对病毒的新型病毒的侦测的解决途径存在的问题我们研发了基于知识图谱的网络安全威胁预警系统，它是一个支持大数据平台的基于知识图谱的网络安全威胁预警系统，既为用户提供高效可靠的大规模数据搜集分析处理能力，又为用户提供可能出现新病毒的预警信息，还有友好而简单的操作界面。本系统采用最适合的国密算法SM2与SM4进行混合数据加密，保障了病毒攻击情况数据和服务器反馈和数据信息的安全性。相较于上述的解决方案，本系统的主要优势如下：

1. 基于flume+kafka+storm的高效大数据平台架构，保证大量攻击情报数

据的即时传输与处理。用户在使用时可以将其部署在服务器上，用于检测服务器提供服务的整个集群的病毒攻击情况并加以分析。如果出现数据库中未知病毒攻击情况则及时更新到病毒库中。

1. 可自学习、自生成的知识图谱与实时更新的自适应识别算法，保障对

于未知病毒的提前预警作用。利用知识图谱，我们使网络安全威胁预警系统具有了知识推理的逻辑结构能力，即可以分析不同可能攻击方式攻击路径之间的关联关系；同时实体、属性、关系能更好的理解语义范围域，使整体结构更加清晰易懂且易扩展，提升上层应用例如搜索和攻击方式匹配的准确率。通过知识图谱的建立，整理出现有病毒的特征和攻击流程，并提前列出大量可能的攻击步骤，每次攻击时记录下攻击步骤的触发情况并记录下一段时间内的频率，当未来某时刻出现了疑似攻击行为但并未产生受攻击警报时，可以通过匹配以往受攻击步骤触发的频率进行匹配，若相似度较高则将其视作新的病毒并纳入病毒库以实现动态扩充病毒库的功能。

（3）可追踪溯源的黑客发现策略。通过分析网络报文段中首部的ip地址分析出攻击来源，如果是肉机发起攻击可以分析肉机方面收到的报文信息通过追踪算法查找到攻击者源ip，减少损失，增大破获攻击案件的成功率。找到源ip后对其发出警告，并联系网络警方加以监督管控。并将从源ip及肉机ip发出的消息皆视为黑名单，以后与其他正常信息区别对待。

（4）采用SM2 和SM4 国密算法混合加密传输数据。当服务器检测到攻击并将攻击数据上传到处理系统、处理系统返回处理消息、处理系统连接数据库传输数据时，本系统随机生成一个SM4 算法的随机密钥key，用SM4 算法对需传送的明文数据加密，然后再用SM2 算法对该密钥key 进行加密。然后再进行点对点分布式网络中完成版权信息和侵权记录的传输，保证了攻击信息传输及分析处理过程的数据安全性。

**1.4 应用前景分析**

在这个超过38亿人使用互联网的全球互联时代，及时发现网络中的新病毒、防止病毒扩散能大大保证用户的信息安全。我们的网络安全预警系统，能基于现有知识库和采集到的实时数据去学习新病毒或变种病毒，及时发现病毒模式作出反馈，用以帮助其他用户防御进行威胁防范。我们的系统较之传统的病毒防御机制，对网络中的病毒反应更加迅速，可拓展性更强。

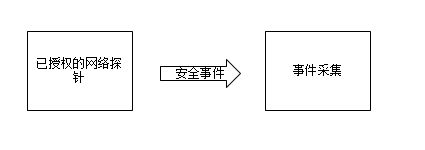
**二、作品设计与实现**

**2.1 系统概述**

我们设计的系统使用场景是：通过将系统部署到类ISP的网络交换中心上，收集网络中各个节点防火墙等系统防护软件发送的安全事件，通过大数据平台对采集到的安全事件进行识别，对识别成功的攻击事件发出告警，对识别失败的事件序列进行学习，在进行统计之后存储数据到数据库中。再通过web应用展示给普通用户或者管理员。

**2.2 系统架构设计**

**2.2.1 系统层次架构设计**

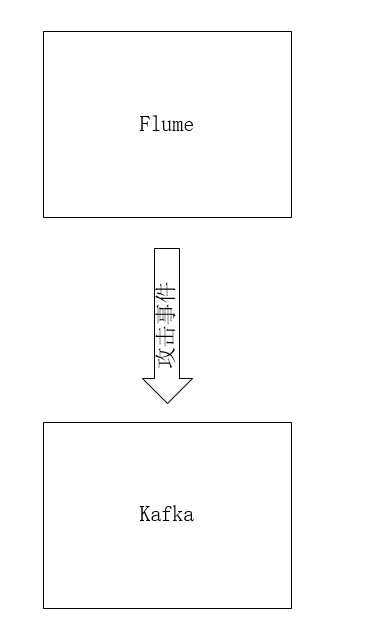


在生产环境中，我们设计系统采集来自网络中各个节点上的防护软件（探针）发送的安全事件信息。在开发环境中，由安全事件模拟器模拟探针作为数据源，在经过授权之后，通过系统提供的网络接口，向系统发送安全事件信息。

**2.2.2 系统模拟攻击架构设计**

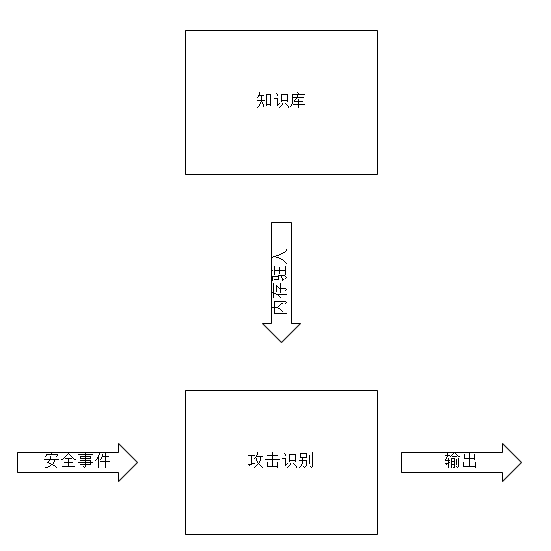


**(1) 事件采集：**



通过Flume提供的接口，收集已授权网络发送的安全事件，再提交到消息中间件Kafka。

**(2) 事件识别：**



a. 知识库驻入内存：在系统启动进行初始化时，访问图数据库Neo4j，将病毒知识库存储到内存中，以避免在对事件做攻击识别时进行频繁的数据库访问，降低系统性能。

b. 攻击识别：基于内存中的病毒知识库，对安全事件进行识别。

**(3) 病毒学习：**

当事件识别失败，而发生安全事件的主机已经故障时，触发病毒学习模块。系统学习历史事件记录，发现变种病毒/新病毒的攻击模式并添加到知识库。

**(4) 主机评估：**

根据对主机危险行为的统计数据，评估主机危险度。

**(5) 数据统计：**

统计攻击成功次数、总攻击次数等数据。

**(6) 数据持久化：**

将安全事件和统计数据存储到数据库

**(7) 通知可视化模块：**

通知可视化模块新的事件已处理，及时展示最新事件。

**2.2.3 系统入侵检测架构设计**

**2.3 系统核心技术**

**2.3.1 集成flume、kafka与storm的大数据平台架构**

在生产环境中，要使系统能及时处理并发的大规模数据，只有部署计算机集群，搭建分布式计算环境系统性能才能满足要求。对大数据平台框架我们做了如下选择：

**（一）信息采集系统：Flume**

考虑到未来系统可能被企业用于监控内部主机状态，或者网络安全公司用于用户保证信息安全等多种应用场景，需要可定制性强的消息采集系统，所以我们选择了Flume。Flume在数据输入上支持：

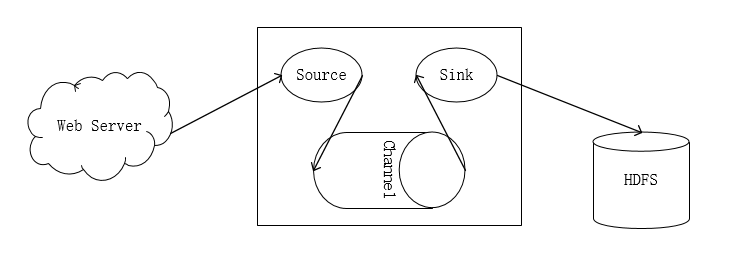
(1) Avro：监听Avro端口来接受来自外部Avro客户端的事件流。利用Avro可以实现多级流动、扇出流、扇入流等效果。

(2) Spooling Directory：监视指定目录，读取新文件的内容作为数据源。

(3) NetCat：监听一个指定的端口，并将接收到的数据的每一行转换为一个事件。

(4) HTTP：基于http协议收集消息。

Flume流动模型：



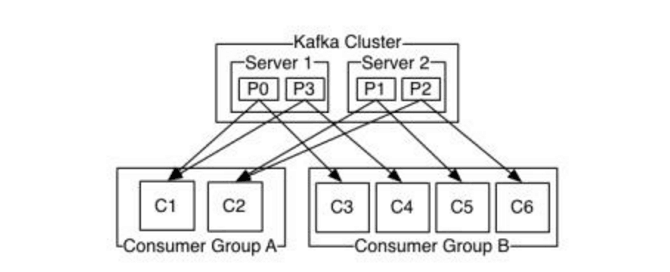
其中Source是Flume公开的数据输入接口，Channel是对数据进行处理的中间件，Sink是根据使用需求设置的输出方式。

Flume允许复杂的数据流动模型，事务性的数据传递保证了数据的可靠性，并且Flume支持文件型Channel，提供了可恢复性。

**（二）消息中间件：Kafka**

考虑到数据源（生产者）产生速度大于事件处理（消费者）的速度，大量数据到来时，当内存不足，或者数据采集模块所在主机故障时，将造成数据丢失。我们需要在事件生产与事件消费之间设计缓存机制，对已到达服务器但还来不及处理的事件进行文件缓存，保证即使主机故障，数据仍然不会丢失。

所以我们选择Apache Kafka作为事件收集与事件处理的中间件。Kafka可以是一台服务器，也可以是服务器集群，统称为broker。每条发布到broker上的消息都有一个类别，成为topic。而每个topic下对应了多个持久化的数据文件，也即partition。每个topic在集群环境中有多个备份。



Kafka对数据进行分布式的、基于文件的持久化，并且通过Zookeeper来管理数据的消费记录等，实现了主机故障后快速恢复的功能，满足分布式系统的需求。

**（三）实时流处理：Storm**

安全态势感知系统实时要求高，使用Apache Hadoop这样的大数据批处理系统是不可行的。只有Storm这样的实时流处理平台能满足实时性要求。Apache Storm具有如下优势：

(1) 低延迟与高性能

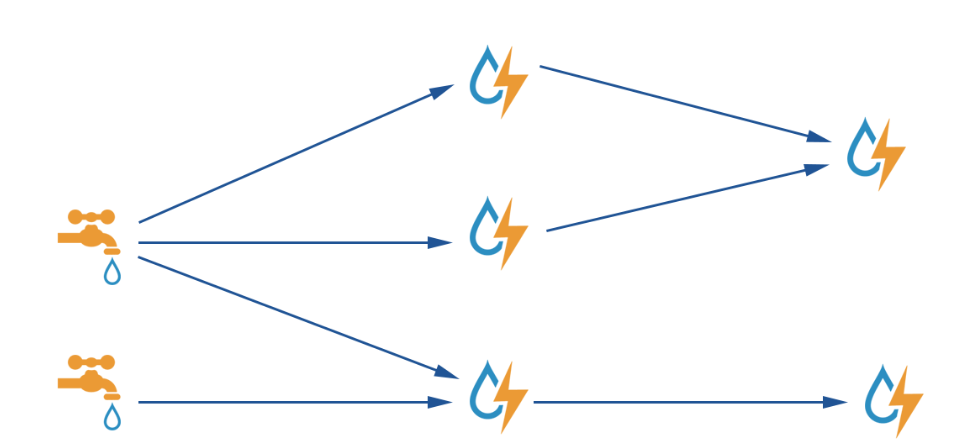
Storm集群主要由工作进程、线程和任务三个实体构成。每台主机上可运行多个工作进程，每个工作进程又可以创建多个线程，每个线程可以执行多个任务，而任务就是真正进行数据处理的实体。针对开发者创建的任务拓扑图，Storm能根据集群中各个主机的性能，合适地分配任务。

(2) 高可靠与高容错

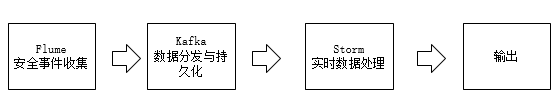
Storm可以保证接受的所有数据都能被完全处理，而如果数据在处理过程中出现了一些异常，Storm会重新部署出错的处理单元，保证处理单元永远运行。

(3) 流处理

批处理框架Hapoop运行的是MapReduce job，当job结束后主机将会进入闲置状态。这对安全事件实时处理是十分不利的。而Storm运行的是拓扑，Storm能不断根据拓扑分发任务到处理单元。

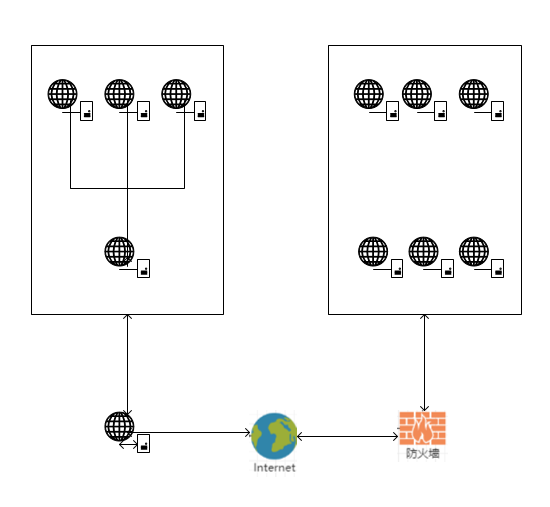


构建好的大数据平台数据处理流程：



**（四）生产环境部署**

生产环境下集群网络结构：



(1) worker1、worker2 ...：我们部署的集群环境，每台主机都部署了Zookeeper、Kafka和Storm；

(2) nimbus：作为leader加入集群，部署了Zookeeper、Kafka、Storm和Flume，用于协调集群工作，提交任务，以及安全事件收集；

(3) 公网代理服务器：使用内网穿透技术将内网中nimbus上Flume公开的数据输入接口，暴露到Internet上，从而允许受监控的主机/网络可以来自任何IP，只要经过授权。

(4) 防火墙：防火墙部署在网络设备上，拥有探针，对网络流量进行监控，并在发生安全事件时，将事件信息提交给态势感知系统。探针等技术实现已超出态势感知系统范围，不作深入说明。

**附：内网穿透实现**

要使得系统能通过Internet采集安全事件，必须将Flume数据输入接口暴露到Internet上。而由于网络交换设备都使用了对IP和端口进行重新映射的策略，处于内网中的主机无法直接通信。

于是我们引入公网代理服务器。使用基于TCP、UDP的开源项目frp（Fast Reverse Proxy）实现Flume数据输入接口的暴露。frp是一个高性能的代理软件，支持HTTP、HTTPS、P2P、加密TCP等多种服务。

**2.3.2 基于国密SM2 和SM4 混合算法的数据传输协议**

此算法用于传输攻击流程中的攻击情报数据以及交互的信息。提高对中间人攻击等网络攻击方式的抵御能力。通过两端的公私密钥，使得数据能够安全有效且高速的传递。

在受攻击记录和受攻击情况的传输过程中，我们利用SM4算法加密速度快和SM2算法加密安全性高、密钥管理简单的特点，使用国密SM2和SM4混合算法对受攻击态势进行加密。首先，随机生成一个SM4 算法的随机密钥key，用SM4 算法对需传送的明文数据加密；然后，使用SM2 算法对该密钥key 进行加密，在保证数据安全的同时又提高了加/解密的速度。

SM2 密钥协商算法与SM4 算法

在本系统中，使用SM4算法加密需要发送的数据明文，SM4算法将明文和密文

均看成4个32比特字，我们表示为：

明文：

密文：

轮密钥：

反序变换：

加密变换：

 。

对方接收到数据时，解密解密变换与加密变换结构相同，不同的仅是轮密钥的使用顺序，在本系统中SM4算法的方法和描述如表：

表 SM4算法主要方法

|  |  |  |
| --- | --- | --- |
| 方法 | 返回值 | 描述 |
| dncryptData\_CBC | String | 使用CBC模式解密 |
| encryptData\_CBC | void | 使用CBC模式加密 |
| decryptData\_ECB | String | 使用ECB模式解密 |
| encryptData\_ECB | String | 使用ECB 分组模式加密 |
| sm4.setSecretKey | void | 设置SM4密钥 |

本系统使用SM4算法对明文数据进行加密，但SM4算法的缺点是消息安全取决于对密钥的保护，泄漏密钥就意味着已加密信息可能被破解。因此，系统同时使用了SM2密钥协商算法对SM4算法生成密钥key进行加密。SM2算法属于非对称密钥算法，使用公钥进行加密，私钥进行解密，已知公钥求私钥在计算上不可行。发送方用接收者的公钥将消息加密成密文，接收方用自已的私钥对收到的密文进行解密还原成原始消息。使用SM2算法的方法及描述见表：

表 SM2算法主要算法

|  |  |  |
| --- | --- | --- |
| 方法 | 返回值 | 描述 |
| encrypt | static byte[] | SM2加密 |
| decrypt | static byte[] | SM2解密 |
| sign | static byte[] | 标签展示 |
| verifySign | Static boolean | 检验标签 |

**2.3.3 基于Neo4j的知识图谱和基于MySQL的数据库平台**

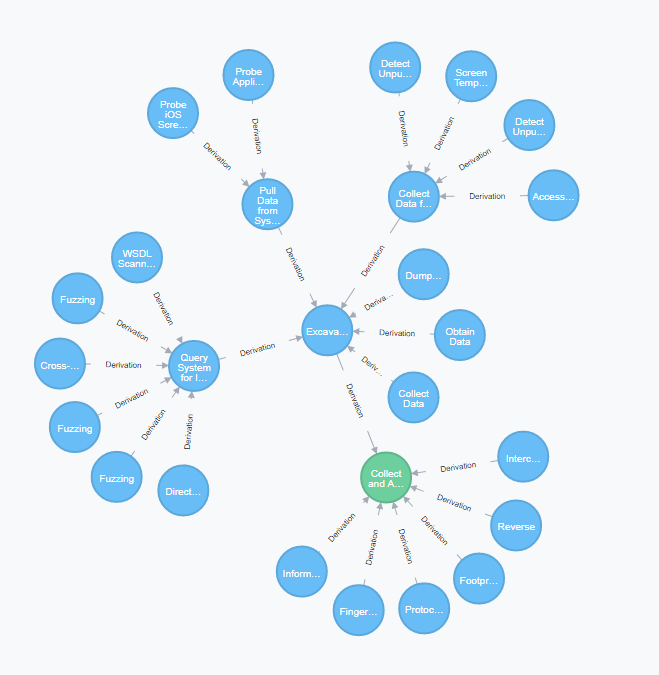
**（一）Neo4j**

Neo4j是一个开源的、无Schema、没有SQL、高性能的图数据库，是一个嵌入式的、基于磁盘的、具备完全的事物特性的Java持久化引擎。

图数据库是以图形结构的形式存储数据的数据库。它以节点、关系和属性的形式存储应用程序的数据，能清楚的表征知识图谱中节点之间复杂的关联。

**1.构建病毒规则集**

我们使用爬虫技术从xxx网站获取到安全事件枚举，并根据其事件之间的关联在Neo4j中构建病毒的规则集。规则集示例如下：

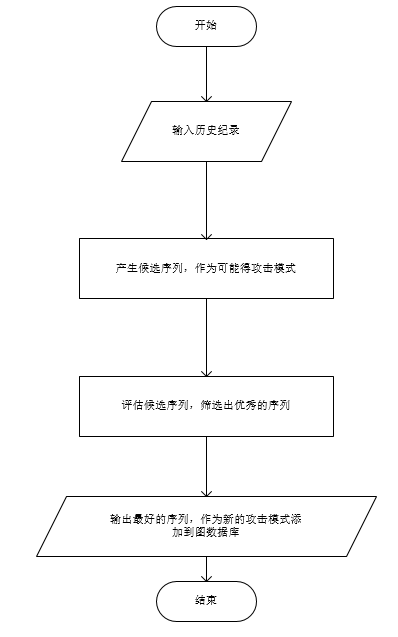


事件节点（蓝色）间建立了Derivation（推导）关系，用以表现事件在一次攻击中的顺序。最后推导至场景节点（绿色），代表一次完整的攻击已经达成，需要系统进行告警。

**2.3.4 实时更新的自适应识别病毒算法**

攻击行为识别模块没有告警，而服务器接收到主机失联的通知时，说明在过去一段时间内有攻击行为发生，而其攻击模式未被记录在图数据库中。这时需要根据历史记录学习出新的攻击模式。

在病毒学习模块被触发时，我们从MySQL数据库中取出一定数量的安全事件记录，再使用学习算法进行处理。学习算法流程图如下：



**1.从历史记录产生候选序列**

候选序列应该满足如下几个条件：

(1) 不包含重复的事件节点

(2) 应当包含一定数目的节点

候选序列生成算法如下：

定义集合V={候选序列}

对原始序列进行一趟遍历，对其中每个节点做如下处理：

遍历所有候选序列，若序列长度为n，如果序列不包含该节点，则将序列按长度1，2

a.如果任何候选序列都不包含节点，则将节点添加到所有候选序列

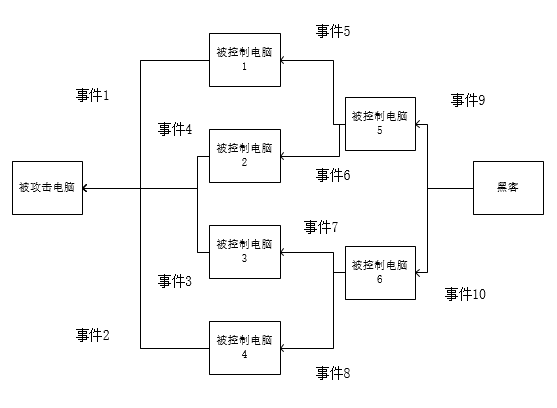
考虑如下的历史记录：

A B A C D F E E H K L ...

对该序列进行一趟遍历，构建

**2.3.5 可追踪溯源的黑客发现策略**

黑客在攻击其他计算机传播病毒时，往往不是通过自己的主机直接发布病毒攻击他人，而是通过控制一定数量的肉机进行攻击来保证自自身的安全和提升传播速度。肉机。肉机又被称为傀儡机，是指被黑客远程控制的计算机。在追寻病毒传播路径的时候，通常很能追踪到黑客本身。而本系统中，采取的黑客发现策略能够在有限层数内，追踪到黑客的原始机。主要采用了对比—剥离的方法策略，计算机中每个交互事件有源IP和目的IP，利用大数据平台的优势，我们将记录下攻击事件的源IP和目的IP。在出现病毒攻击后，触发追踪方法，查询这个知识图谱一系列事件的源IP，然后在根据多源IP查询全库的目的IP，将多台受攻击的计算机的事件源IP不断进行比较，剥离出发送攻击事件的源IP，然后一步一步定位到一个IP，即黑客



攻击层数示意图

如图所示，当我们的被攻击电脑瘫痪后，马上查询事件1-4的来源，得到被控制电脑1-4的IP。再以他们的IP作为目的IP查询到事件5-8，对比事件的源IP地址得到更近一层的傀儡机5-8的IP地址，依次向上追踪最终查找一切的源IP，即黑客。

**2.4 系统功能设计**

**2.4.1 用户模块**

**2.4.1.1 登陆功能**

**2.4.1.2 注册功能**

**2.4.1.3 修改信息功能**

**2.4.2 权限模块**

**2.4.2.1 普通用户权限**

**2.4.2.2 管理员权限**

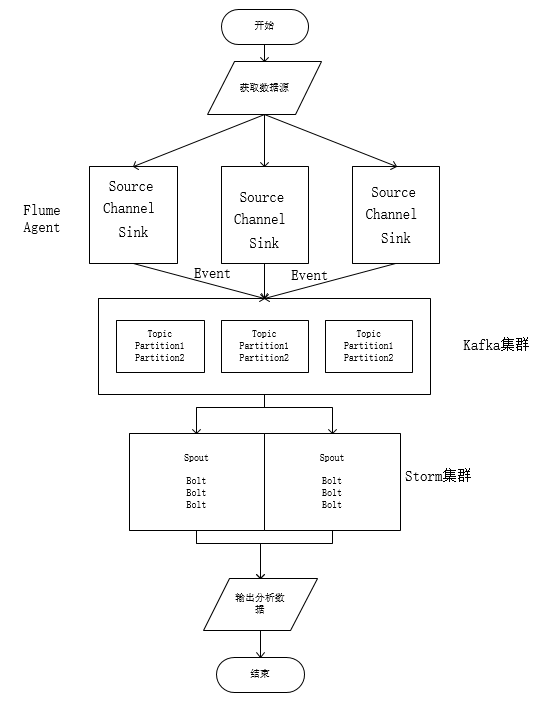
**2.4.3 病毒防御模块**

**2.4.3.1 勒索病毒检测**

**2.4.3.2 建立基于知识图谱的安全规则库**

**2.4.4 大数据平台实时事件处理**

系统引入了大数据平台做实时的事件处理，能够在数据量极大的情况下仍然保持着对数据的实时处理的功能。采用了流行的flume+kafka+storm作为数据平台的构架。系统中，flume从管理下的数以万计的计算机的防火墙和监控软件中获取数据，作为生产者，将数据可靠地发送给消息队列处理系统kafka；kafka作为中间人,解决生产消费速率不匹配问题；storm则充当消费者实时地处理数据。



可从图中得知flume、kafka和storm三个系统都有自己独立的一套结构和功能，同时三个系统能很好地合作连接到一块来作为一个整体。此大数据平台中的数据流向在图中能很好地体现出来，具体如下：

**1．数据源**

每台计算机中的防火墙能保护脆弱的服务，控制对系统的访问，它几乎对计算机所有的与外界的网络往来就有着严密的记录。因此我们把每台计算机的防火墙作为数据源，能够准确高效的获得计算机的网络往来记录。同时，监控软件也作为了我们获取数据的数据源，帮助信息更完善和准确。

**2.数据流向flume**

每个flume agent从数据源不断地获取数据，得到数据后将数据暂时存储在channel中，在sink中将数据分为一条条的基本单位event发向kafka集群中。Flume处理的数据具有可靠性。

**3.event到达kafka集群**

数据被作为event推向kafka后，根据不同的类型被分往不同的topic中，并储存在物理内存中。物理内存中以partition作为单位存储，等待storm来消费。传输时每条数据拥有自己的offset，并采用相应的机制来保证了有序性和传输可靠性。

**4.storm‘消费’**

Storm从Kafka中消费到信息后。其中的组件Nimbus则会根据当前系统的繁忙情况和数据量，将处理信息的作业分给相对空闲的Bolt进行处理。保证了系统的高效运行和负载均衡。

该系统的实现策略和采用的技术很好的解决了需求。数据来源全面可靠、flume的可靠性和可恢复性、Kafka的有序性以及storm的高效性和负载均衡性是该系统大数据实时处理功能的基础和关键。

**2.4.5 病毒发现和自增加策略**

一个好的病毒感知系统应当对新病毒的识别功能，本系统除了能识别新病毒以外，还能自动更新病毒库，将病毒知识图谱做分支扩展或者创建一个新的知识图谱。此功能的实现主要依赖于大数据平台的计算能力和采用的知识图谱的扩展能力。知识图谱从根本上来讲是一个图，比起关系型数据库，实现知识图谱的非关系型数据库更能表达出病毒每个行为之间的连接以及下个行为趋向。新病毒的出现会导致在对原病毒知识图谱进行遍历对比时出现终止的情况，但最终却到达了最终结点即电脑中毒了，这就说明出现了新病毒。我们将已保存的新病毒行为路径进行分析，与从其他电脑上的获取的此新病毒的行为特征进行比对精简，当我们的比对数目达到一定量后，便能对新病毒的行为特征和路径准确掌握。至此，我们往我们的病毒库中添加图谱分支或者一张新图。此功能中包含了一些机器学习的含义，并是自动完成的，我们所构建的算法的流程图如下：

**2.4.6 黑客发现策略（重新画图）**

黑客发现和黑名单是该系统的一大创新点功能，能在有限的攻击层数范围之内追根溯源，有效的辨别出黑客控制的肉机，发现此病毒的始作俑者。此功能算法主要采取递归查询的方法策略，即以受攻击的电脑的ip作为目的IP地址出发查询发布攻击事件的傀儡计算机，并通过其他若干台的计算机所查询出的傀儡机进行IP比对，判断出此傀儡机的传播影响度，在适当的时候将此计算机的IP加入到黑名单中。加入黑名单的IP将会被广播或者被屏蔽，可终结它对病毒的进一步传播。同时，再以此肉机的IP为目的IP地址查到源IP地址，不断递归向上向上，最终得到一切的源头，即黑客。

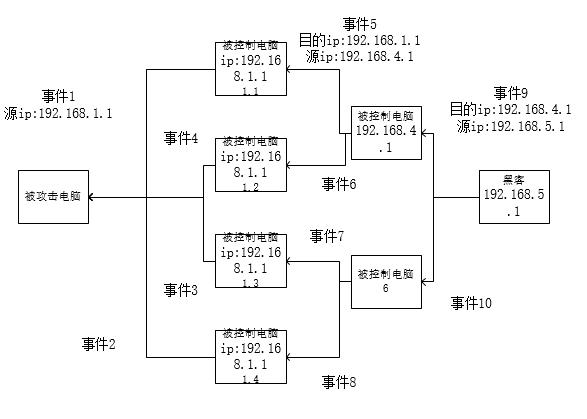
黑客发现和黑名单功能的实现主要是利用大数据平台和采集到的数据，根据一定的算法进行计算，从而找到黑客和实现黑名单。算法流程图如下：



攻击层数示意图

如图所示，当我们发现某台电脑被病毒入侵，马上根据采集到的数据进行查询，即以该电脑的IP为目的IP查询出病毒事件的源IP，并标记此源IP。判断此IP是否是所有病毒事件传播的最终源IP，如果是，此源IP则为黑客所在。若不是，继续判断该IP是否被标记到达了一定次数N，加入黑名单，对加入黑名单的IP采取一定措施。并继续以此IP为目的IP进行查询，直到到达顶层，找到黑客。

下面对一个该功能案例进行说明：



递归查询流程图

如图所示，当我们的电脑被攻击瘫痪后，立刻查询我们的病毒知识图谱，查出攻击事件1-4，查到发送事件1-4的电脑源IP192.168.1.1、192.168.1.2、192.168.1.3和192.168.1.4。同时，通过我们对其他受攻击电脑的查询显示IP为192.168.1.1的电脑在大范围的发送病毒事件，此时已经达到了一定次数，我们将其加入黑名单，以杜绝它的继续传播。我们继续查询我们的病毒知识图谱，得到攻击事件5-8，以查到的源IP地址，以查到的源IP作为目的IP查询出发送事件5-8的电脑的源IP地址，可得到电脑6的IP和IP地址192.168.4.1。同理，如此不断递归迭代下去，越来越多的傀儡机会被加入黑名单，最终黑客地址也会被找到。此功能需要大量的数据和查询计算，大数据平台为我们提供了良好的保障。

**2.4.8 安全态势可视化展示**

**三、作品测试与分析**

**3.1 测试方案**

**3.2 测试环境及测试设备**

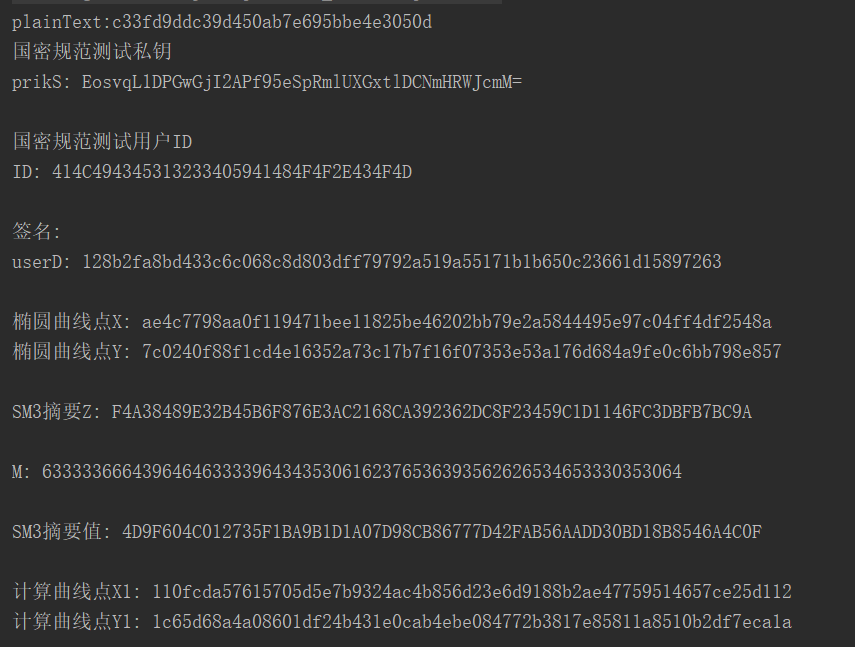
**3.3 算法测试**

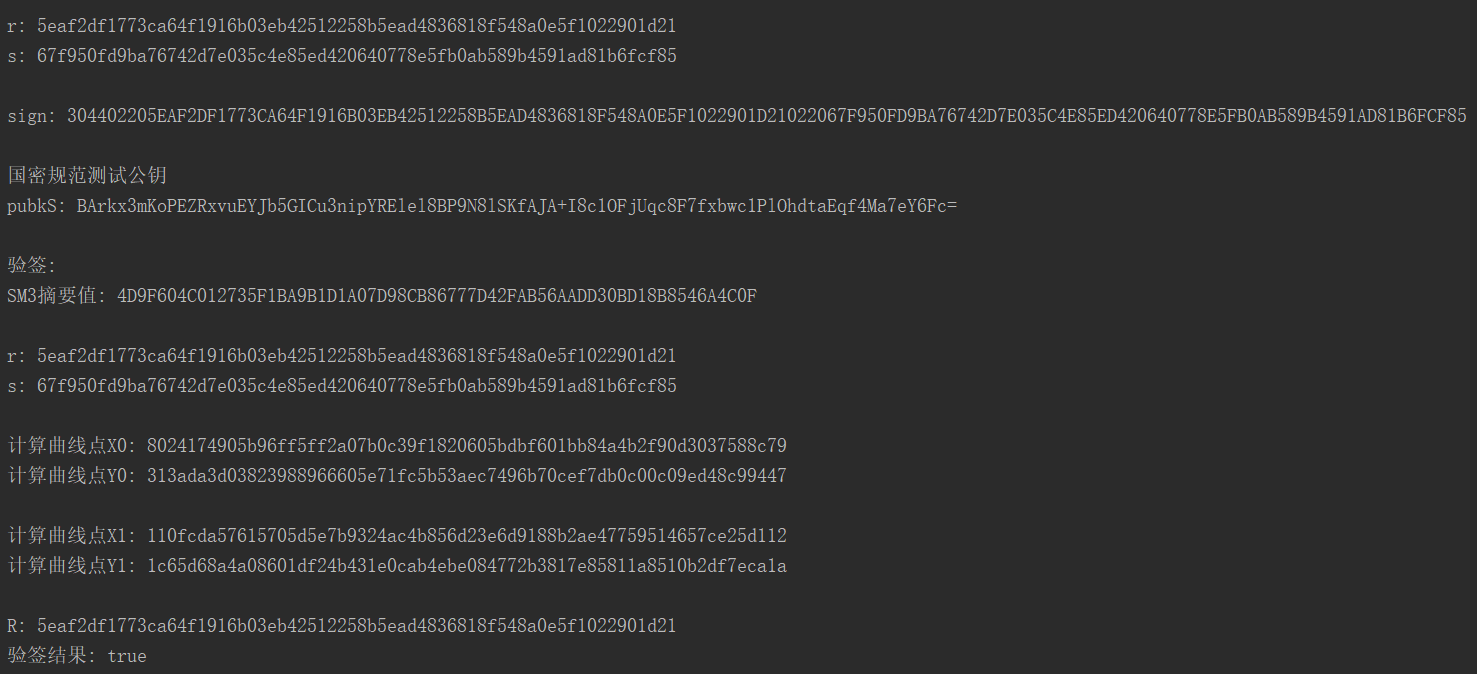
国密SM2 算法的测试用例如表3-2，输入数据来源于国密规范测试私钥和国密规范测试公钥以及随手输入的较长字符串。

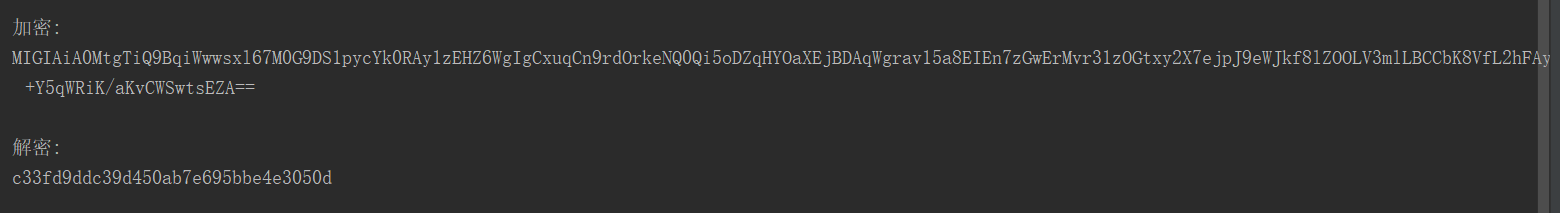
**3.3.1 国密SM2 算法测试**

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | No.1 | 模块名称 | SM2算法 |
| 测试方法 | 黑盒测试 | 测试日期 | 2018.5.11 |
| 测试说明 | 根据国密规范测试私钥和国密规范测试公钥，判断SM2加解密算法能否正常运行及实现密钥协商 | | |
| 预置条件 | 系统将要进行数据传输 | | |
| 判断准则 | 输入的字符串为随机输入，可多次重复实验，对比加解密后字符串数据是否一致。 | | |
| 测试输入 | plainText:c33fd9ddc39d450ab7e695bbe4e3050d  国密规范测试私钥  prikS: EosvqL1DPGwGjI2APf95eSpRmlUXGxtlDCNmHRWJcmM=  国密规范测试公钥  pubkS: BArkx3mKoPEZRxvuEYJb5GICu3nipYRElel8BP9N8lSKfAJA+I8c1OFjUqc8F7fxbwc1PlOhdtaEqf4Ma7eY6Fc= | | |
| 测试输出 | sign: 3044022040F1EC59F793D9F49E09DCEF49130D4194F79FB1EED2CAA55BACDB49C4E755D102206FC6DAC32C5D5CF10C77DFB20F7C2EB667A457872FB09EC56327A67EC7DEEBE7  R: 40f1ec59f793d9f49e09dcef49130d4194f79fb1eed2caa55bacdb49c4e755d1  验签结果: true  加密:  MHYCIAR9YwJMudf/t74d5wYwKjY/dS5p7RO3B6cy0kZ/XAdcAiBDVSSrzpzPdwnDzdk3vDMU6euXtEo68i6g/P0os5MhxwQgrgWjwx1o8lmMh74ec1TxPs+gL34ZOmTAgZ+0S2Vgr2UEDlV8ogXiOt2mXJRR2UFO  解密:c33fd9ddc39d450ab7e695bbe4e3050d | | |
| 测试评价 | 测试通过，SM2 算法能够正常运行 | | |

测试结果如图







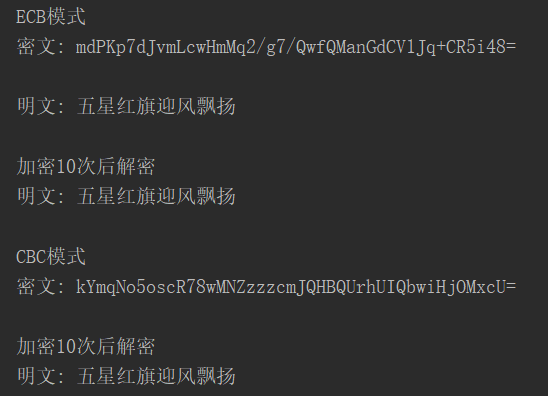
发送方和接受方在数据传输前进行密钥协商，结果如图所示，与预期结果一致，表明通过SM2 算法测试。

**3.3.2 国密SM4 算法测试**

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | No.2 | 模块名称 | SM4算法 |
| 测试方法 | 黑盒测试 | 测试日期 | 2018.5.11 |
| 测试说明 | 通过预设的公钥和输入数据，分别加密1 次和10次，再进行解密 | | |
| 预置条件 | 系统将要进行数据传输 | | |
| 判断准则 | 使用SM4 算法加解密，与输入的明文进行对比，判断是否一致 | | |
| 测试输入 | 明文：五星红旗迎风飘扬  密钥：JeF8UygsiOMfs2Y8 | | |
| 测试输出 | CBC模式  加密1 次后解密明文：五星红旗迎风飘扬  加密10次后解密明文：五星红旗迎风飘扬  ECB模式：  加密1 次后解密明文：五星红旗迎风飘扬  加密10次后解密明文：五星红旗迎风飘扬 | | |
| 测试评价 | 测试通过，SM4 算法实现了数据稳定地加解密 | | |

测试SM4 算法时的加解密状态，通过国家密码局推荐的公钥以及自定义的一句话作为加密数据，分别加密1 次和10 次，最后进行解密。

得到的结果都是输入数据，表现出SM4 算法的正确性与稳定性。

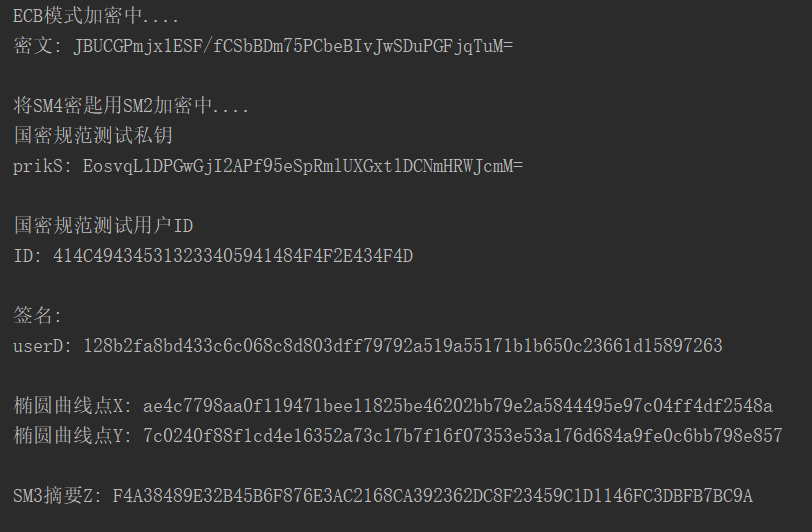


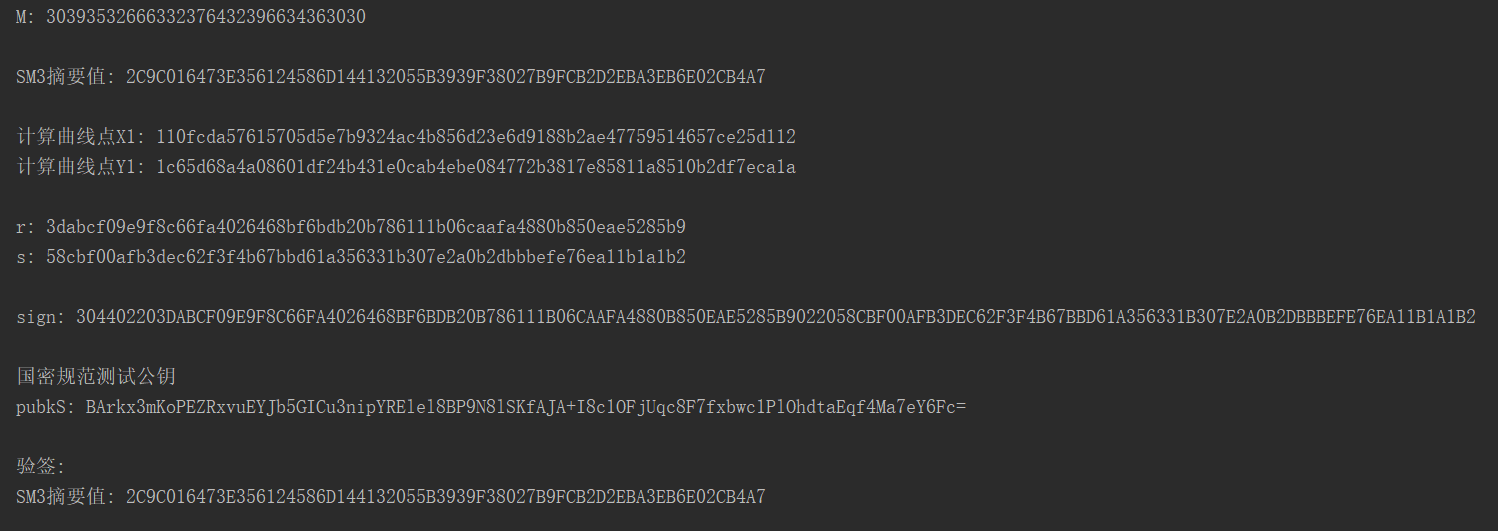
**3.3.3 基于国密SM2 和SM4 混合算法的数据传输协议测试**

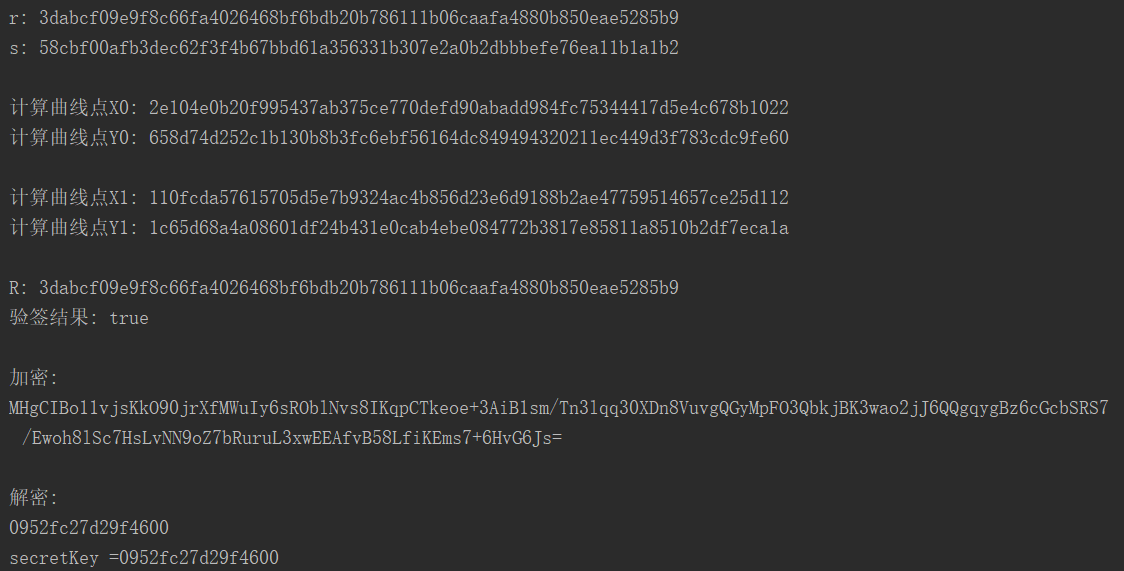
测试SM2 和SM4 混合算法时，选取设计好的版权信息区块链的数据作为输入数据，使用SM4 算法加密数据明文，SM2 算法加密密钥，如下表是SM2 和SM4 混合算法的测试用例：

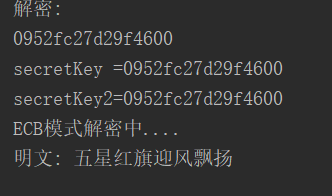
|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | No.3 | 模块名称 | SM2 和SM4 混合算  法 |
| 测试方法 | 黑盒测试 | 测试日期 | 2018.5.11 |
| 测试说明 | 使用SM2 和SM4 混合算法对版权信息区块进行加解密 | | |
| 预置条件 | 在进行攻击情报传递时，需要对情报内容数据加密 | | |
| 判断准则 | 对比加密前和解密后数据的一致性判断算法的正确性 | | |
| 测试输入 | 数据明文：五星红旗迎风飘扬  国密规范测试私钥  prikS: EosvqL1DPGwGjI2APf95eSpRmlUXGxtlDCNmHRWJcmM=  国密规范测试公钥  pubkS: BArkx3mKoPEZRxvuEYJb5GICu3nipYRElel8BP9N8lSKfAJA+I8c1OFjUqc8F7fxbwc1PlOhdtaEqf4Ma7eY6Fc= | | |
| 测试输出 | 验签结果: true  secretKey =dccfec8a27854208  secretKey2=dccfec8a27854208  解密出明文: 五星红旗迎风飘扬 | | |
| 测试评价 | 测试通过，版权信息区块数据明文经过SM2 和SM4 混合算法加解密后数据未发生变化。 | | |

下图是测试SM2 和SM4 混合算法的输出结果。









**3.3.4 未知病毒发现算法测试**

**3.3.5 红黑树病毒匹配算法测试**

**3.4 功能测试**

**3.4.1 未知病毒检测模块**

**3.4.2 攻击者追踪模块**

**3.4.3 实时受攻击情况展示模块**

**3.4.4 数据传输安全性测试**

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | No.3 | 模块名称 | 数据传输安全性测试 |
| 测试方法 | 黑盒测试 | 测试日期 | 2018.5. |
| 测试说明 |  | | |
| 预置条件 |  | | |
| 判断准则 |  | | |
| 测试输入 |  | | |
| 测试输出 |  | | |
| 测试评价 |  | | |

**3.5 性能测试**

**3.5.1 安全攻击负载测试**

**3.5.2 可视化测试**

**四、创新性说明**

**4.1 高效的大数据平台架构**

基于知识图谱的网络安全威胁预警系统中所搭建的大数据平台是重要的一个组成部分，是系统的一个基本点所在，也是系统的一个创新点。大数据平台用来解决病毒感知有着他独特的优势。大数据平台是实现病毒的检索、预警等基本功能的基本，也使得我们实现自学习的、自生成的知识图谱和黑客发现等创新功能的构想得以变成现实。并且，大数据平台的很多特性成就了此系统的很多特性，它使得系统高效、快速、及时、准确。大数据平台的利用是本系统的一大创新！

**4.2 可自学习、自生成的知识图谱**

可自学习、自生成的知识图谱是本病毒感知系统中最为创新的一点。传统的病毒检测方法包括了特征代码法、校验和法、行为监测法、软件模拟法。这些方法依据的原理不同，实现时所需的开销不同、检测范围不同，各有所长，也各有自己的一些缺陷。例如：特征代码法无法识别新病毒、不能检测多形态病毒；校验和法往往会发生误报警，不能报出病毒名字；行为监测法可能会误报警、实现有一定难度。我们使用的基于知识图谱的检测本身上属于行为监测法，但他完成去除了行为监测法的缺点。知识图谱是对病毒行为特征的构建成的具有多个行为多条路径的复杂图，但最终的结果唯一。能十分恰当的描述出病毒的行为，十分复杂的多形态病毒也不例外。在对病毒进行检测时，运用当前已经十分成熟的图论的相关算法进行遍历，由于本身采用的非关系型数据库，检测速度十分的快，准确性十分的高。并且，采用的知识图谱还具有自学习、自生成的特性，对新病毒的处理也毫不逊色。通过我们的新病毒发现和自生成的算法，在发觉了新病毒的出现后，新病毒的行为会被加入到原来已有图的分支节点或者一张新的图谱当中，并在之后的检测中，对当前所新构建的图谱进行不断的修正，直到达到了最精确为止。知识图谱的采用还有许多的优点，在此不再一一阐述。总之，知识图谱让病毒的发现变得更容易、更快速、更全面！

**4.3 可追踪溯源的黑客发现策略**

病毒的防护和预警是一个安全系统的基本功能，但也是基础功能。如果能直至根源地解决问题，无疑是一个更深层面的解决方案。黑客往往躲在黑暗中，把自己的主机隐藏在层层的傀儡机后面，本系统中提出的可追踪溯源的黑客发现策略便是一个解决根源的策略，系统通过不断的递归查询和比对，一步一步地识别到傀儡机，顺藤摸瓜，一层一层地往上梳理，最终达到顶层，解决病毒传播的源头来达到防护目的，无疑是病毒防护系统的一个创新。

**4.4 B/S架构的通信安全**

在基于知识图谱的网络安全威胁预警系统中数据的传输的安全性问题有国密SM4算法提供保护。相较于其他国际算法，如MD5,DES或者RSA，首先，使用国家密码局发布的算法更加符合国情，对于国人来说更加自主可控，安全性更高，国密SM4算法性能更加优良；其次，将国密SM4算法应用于大数据服务器的加解密中，是的数据传输过程更加安全可靠，防止内容被恶意篡改。在攻击记录和受攻击情况的传播过程中，考虑到SM4算法加密速度快和SM2算法加密安全性搞、密钥管理简单的优点，对区块中数据使用国密SM2和SM4混合算法进行加密，国密SM4算法使用SM2算法动态协商的密钥进行数据的加密，所以加密数据的安全性就依赖于国密SM2协商密钥算法的安全性，国密SM4利用分组加密的形式，在传输信道中传输效率更高，速度更快。

**4.5实时更新的自适应识别算法**

**五、总结**

参考文献

[1]基于数据挖掘的网络安全态势分析 姚钦锋 2012.06