# 大数据环境下威胁的协作式检测综述

张俭鸽 郭渊博 马 骏 陈 越
(信息工程大学数学工程与先进计算国家重点实验室 郑州 450001)

**摘 要** 恶意的不法分子采用直接或间接的方法攻击个人、机构、国家,从而使其遭受不同程度的威胁。此类信息的形式多种多样,数据量巨大,而且需要被高速地处理。因此,首先对5种典型的协作式检测模型Esper,Hadoop,Agilis,Storm和Spark进行分析、比较,阐述不同模型所适用的网络环境;然后对网络环境中常用的攻击手段DDoS,MITM,APT进行分析,说明检测这些攻击适合采用的模型;最后给出威胁的协作式检测架构模型部署方案,该方案包括发送和接收处理两个组件,并指出可根据实际需要进行不同模型的架构部署;特别地,给出了对等网络、分等级的安全域网络、分层结构网络中架构模型的部署方案。

**关键词** 威胁,协作式检测,攻击,架构模型,大数据
**中图法分类号** TN915.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.10.003

## Review of Collaborative Detection of Threat in Big Data

ZHANG Jian-ge GUO Yuan-bo MA Jun CHEN Yue
(State Key Laboratory of Mathematical Engineering and Advanced Computing, The PLA Information Engineering University, Zhengzhou 450001, China)

**Abstract** Some malicious and illegal persons take advantage of direct or indirect methods to attack some person, organization and nation, so that they suffer from different degrees of threats. The type of information is various, volume of data is large and it needs to be processed at high speed. Therefore, we firstly analyzed five typical collaborative detection models which are Esper model, Hadoop model, Agilis model, Storm model and Spark model. Moreover, we made comparison of them and expatiated the network environment for different models. Then, we analyzed common attack methods in the network which are DDoS attack, MITM attack and APT attack, and explained detection models for these attacks. Finally, we provided the deployment scheme of collaborative detection of architecture model for threats. The scheme includes two components which are sending component and receiving processing component. Then we pointed out that the architecture of different models can be deployed according to practical requirements. Especially, we provided the deployment scheme of architecture model in peer to peer network, ranked security domain network, and hierarchical structure network.

**Keywords** Threat, Collaborative detection, Attack, Architecture model, Big data

## 1 引言

网络的开放性使得用户的上网行为会被互联网记录,同样,攻击者的攻击行为也会在网络上留下踪迹。这样,互联网就为星罗棋布的用户活动铺就了道路。特别地,在大数据时代,各种各样的踪迹数据分布式地存储在网络的多个节点上,每时每刻都会产生海量的数据,在这些海量的数据中,大部分都是正常的业务活动。然而,除了正常的业务活动之外,还有一些活动可能是某些蓄意攻击的行为。

2015年5月11日20点40分左右,网易因骨干网络遭受攻击,导致网易旗下的部分服务暂时无法正常使用,直接经济损失超过1500万元人民币。虽然任何一个机构在提供服务时都会采取安全、可靠的防护措施,但是由于网络的开放性,其不可避免地会遭受黑客或恶意人员的攻击。特别地,无论像银行、电信这些传统的金融业务,还是余额宝、招财宝这些新兴的金融业务,都可能是恶意攻击者乐于实施攻击的重点目标。攻击者可能在金融业务运转的过程中采取各种攻击手段对金融业务活动实施威胁。而网络空间攻击对任何机构都会造成严重的后果:收益丢失、名声损坏、信息系统损坏、私有数据或客户敏感数据盗窃[1,2]。

因此,将这些蓄意破坏的恶意攻击活动从正常的业务活动中提取出来,对其报警并及时地采取合理的防御,将在很大程度上降低安全风险和减少经济损失。这就需要对网络上包含正常的业务活动、异常的蓄意攻击活动在内的所有活动协作式地进行实时或定时的分析、关联、预警、防御。于是,在大数据环境下进行协作式的威胁检测技术受到了广泛的重视,

到稿日期:2015-09-16 返修日期:2015-12-07 本文受国家自然科学基金项目(61201220,61309018),国家973计划项目(2012CB315901),十二五预研项目资助。

张俭鸽(1980—),女,博士生,讲师,主要研究方向为网络信息安全,E-mail:jiangezh@126.com;郭渊博(1975—),男,博士,教授,主要研究方向为网络信息安全、态势感知;马 骏(1981—),男,博士后,讲师,主要研究方向为网络信息安全、态势感知;陈 越(1965—),男,博士,教授,主要研究方向为网络信息安全、先进计算。

出现了一些协作式威胁检测系统。目前，Giuseppe A D L 等人提出了一种集中式的在线实时分析的复杂事件处理模型 Esper[3,4]；Dong Cutting 等人提出了一种分布式的批处理模型 Hadoop[5]；Leonardo Aniello 和 Roberto Baldoni 等人提出了一种分布式的在线实时分析模型 Agilis[6]；Nathan Marz 等人提出了一种分布式的、容错的在线实时分析模型 Storm[7]；Michael Armbrust 等人提出了一种分布式的多迭代批处理模型 Spark[8]。

本文首先简述了在大数据环境下 5 种典型的协作式检测模型，对这些模型进行了整体比较，并阐述了不同模型所适用的网络环境；然后描述了网络环境中常用的攻击手段，说明检测这些攻击适合采用的模型；最后给出了威胁的协作式检测架构模型部署方案。

## 2 协作式检测模型

随着大数据技术的发展，出现了很多大数据处理模型和工具，典型模型有 Esper，Hadoop，Agilis，Storm，Spark 等。这些大数据模型的出现为网络和主机的威胁检测提供了较好的技术支持。

### 2.1 集中式在线实时分析模型 Esper

Esper[3]是可对事件进行实时分析并做出反应的模型，如图 1 所示。

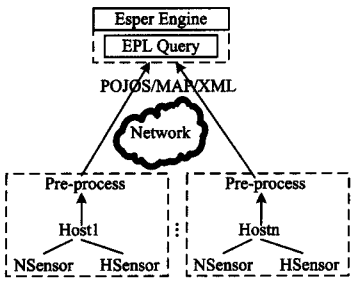


图 1 Esper 模型

由图 1 可知，部署在主机上的网络探针和主机业务探针获取相应信息并对其进行预处理，采用 POJOS(Plain Old Java Objects)、MAP、XML 3 种方式中的一种发送给 Esper 引擎。

Esper 引擎采用内存数据库方式，通过事件处理语言(Event Processing Language，EPL)查询并对复杂事件进行处理。当 EPL 查询匹配出异常信息时，更新全局数据结构并记录恶意 IP 地址，其流程如图 2 所示。

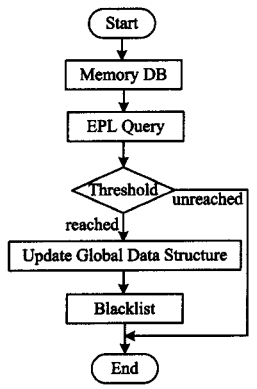


图 2 Esper 引擎流程图

Esper 模型的特点如下。

1)集中式；

2)采用内存数据库，与传统的关系数据库相比，有更好的查询性能，更适合于复杂事件处理(Complex Event Processing，CEP)应用；

3)提供两种机制来处理事件：通过状态机来实现基于表达式的事件模式匹配，这种事件处理的方法是匹配期望存在的事件、不存在的事件或事件的组合；通过 EPL 语句来实现事件流查询，这种事件处理的方法提供了过滤、滑动窗口、聚合、连接和分析等函数，EPL 采用视图将构造的数据放入到一个事件流中并用其驱动数据的流动，在数据流动的过程中对数据进行处理，得到最后需要的结果；

4)每秒能处理 50 万条事件，输入信息流速度能达到 70Mb/s；

5)当面临新的威胁时，能够通过集成/删除 SQL 查询语句来动态调整其检测逻辑；

6)开销低。

根据上述特点，Esper 模型常用在股票系统、风险监控系统、业务活动监控系统、欺诈检查系统、入侵检测系统等对实时性要求比较高的系统中。然而，Esper 模型是集中式的，比较适合在小规模网络中进行部署。

### 2.2 分布式批处理模型 Hadoop

Hadoop[5]是一种分布式的、批处理的、基于任务调度的模型。它使用磁盘作为中间交换的介质，数据积攒一批之后由作业管理系统启动任务，Job Tracker 计算任务分配，Task Tacker 启动相关的运算进程；运算结果写入到 HDFS 中，Reduce 任务对通过网络传输到本节点的数据进行运算；整个运算结束后将结果批量导入到结果集中。Hadoop 模型在多节点集群中的部署如图 3 所示。

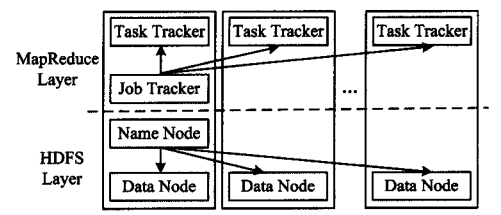


图 3 多节点集群的 Hadoop 模型

由图 3 可知，在 Hadoop 中，用于执行 MapReduce 任务的主机角色有两个：1) Job Tracker；2) Task Tracker。Job Tracker 用于调度工作，Task Tracker 用于执行工作，并且一个 Hadoop 集群中只有一台 Job Tracker。每个 MapReduce 任务都被初始化为一个 Job，每个 Job 又可以分为 Map 阶段和 Reduce 阶段，这两个阶段分别由用户定义的 Map 方法和 Reduce 方法进行处理。在 MapReduce 中，每个 Job 对数据的处理流程如图 4 所示。

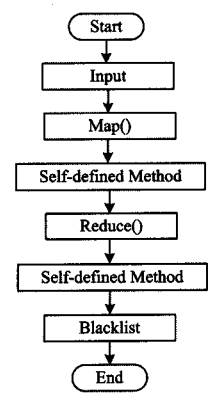


图 4 MapReduce 的数据处理流程

Hadoop 模型的特点如下：

1)分布式；

2)扩容能力，能可靠地存储和处理千兆字节(PB)数据；

3)成本低，能通过普通机器组成的服务器群来分发以及处理数据，这些服务器群总计可达数千个节点；

4)高效率，能在节点之间动态地移动数据，并保证各个节点的动态平衡，使得处理非常快速；

5)可靠性，能自动地维护数据的多份复制，并且在任务失败后能自动地重新部署计算任务。

6)计算逻辑部分的源代码需要自实现，支持各种编程语言，如 Java，Ruby 和 Python 等。

Hadoop 在进行数据分析时，由于需要对数据进行积攒，因此不适合低延迟数据访问的应用，常用在互联网搜索引擎、广告分析、电子商务、网络安全、医疗保健等实时性要求不高的系统中。

### 2.3 分布式在线实时分析模型 Agilis

Agilis[6]是一种分布式的在线实时分析模型，建立在开源 Hadoop 的 MapReduce 的基础设施之上，为了提高反应速度和降低管理费用，增加了一个基于 RAM 的数据存储——IBM Web 软件平台 WebSphere eXtreme Scale，简称 WXS。WXS 提供了关于 Map 的抽象，由多个有关的数据记录组成一个 Map，每一个 WXS Map 都可被分割成多个 partitions，每一个 partition 被分配到一个单独的 WXS 容器中。

在 Agilis 中，所有输入数据一起被当作一个单一逻辑 WXS Map，然后对其进行分割，使得驻留在第一个 Agilis 站点上的每一个 Agilis 节点负责存储本地事件数据组成的 partition；随后，每一个 partition 被映射到一个单个的 Hadoop 输入 split，并被一个 Map Task 实例所处理。

Agilis 模型的组件如图 5 所示。

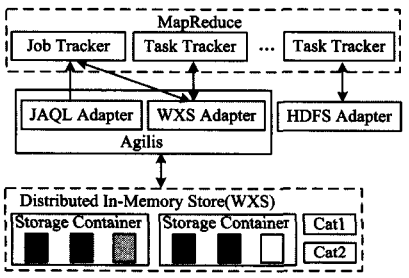


图 5 Agilis 模型

由图 5 可知，Agilis 模型比 Hadoop 模型还增加了一个指定事件处理逻辑的 JAQL[9]数据查询。JAQL 是类 SQL 语言，是专门为 JSON(JavaScript Object Notation)设计的查询语言，且支持流式处理。数据查询交由 JAQL 前端进行处理，JAQL 前端将其分割成多个 MapReduce Jobs，然后在 Hadoop 基础设施上执行。由于 Map tasks 之间的无关性，MapReduce 框架本身就能够很好地支持协作式处理，也允许将 Map tasks 放置在各自靠近其输入数据源的地方。由 JAQL 前端提交的每一个 MapReduce Job 是由多个被封装在 Java 解释性代码中的原始查询的多个 partition 组成。

利用 Agilis 模型进行威胁的协作式检测时，需要监听每一个 Agilis 站点中的 TCP 数据包流，识别不完全连接和失效连接，检测出由可疑源 IP 地址发出的探测包，并将所有生成的检测信息存储在同位的数据存储器中。

JAQL 查询时的数据流程如图 6 所示。

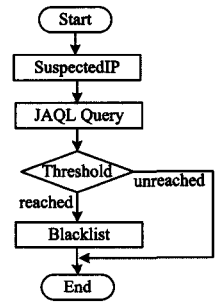


图 6 JAQL 查询的数据流程

Agilis 模型的特点如下：

1)分布式；

2)具备多样式数据分析的能力，比如通过从历史数据中提取模型进行在线处理校准；

3)具有利用一个高级类 SQL 语言 JAQL 指定事件处理逻辑的能力；

4)具有可扩展的性能，数据分割通过 MapReduce 并行化处理数据，避免了基于 RAM 的存储系统的磁盘 I/O 瓶颈；

5)具有通过复制的容错能力；

6)利用开源组件，简化代码验证和认证，提高了可移植性和互操作性，并降低了与现有的分布式事件处理系统相关的复杂性和管理成本。

Agilis 在小规模网络中进行部署时，其性能比较低，当朝着更大规模的网络进行部署时，其相关性能将大幅提高。

### 2.4 分布式容错实时分析模型 Storm

Storm[7]是一种分布式的、容错的、在线实时的数据流分析模型，与 Hadoop 类似，Storm 也可以处理大批量的数据。而且，Storm 在保证高可靠性的前提下还可以让处理更加实时。

Storm 集群主要由一个主节点和一群工作节点(Worker Node)组成，通过 Zookeeper[10]服务进行协调。主节点通常运行一个后台程序 Nimbus，类似 Hadoop 中的 Job Tracker，用于响应分布在集群中的节点，进行任务分配和监测故障。工作节点也运行一个后台程序 Supervisor，用于接收指派并按要求运行工作进程。Nimbus 和 Supervisor 之间的协调是通过 Zookeeper 服务完成的。Storm 集群模型如图 7 所示。

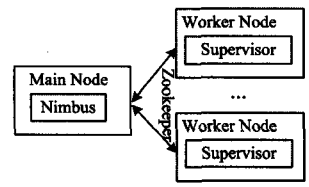


图 7 Storm 集群模型

由图 7 可知，每个工作节点对事件实时处理的逻辑被封装进 Storm 的 topology 中。topology 是一组由 Spouts(数据源)和 Bolts(数据操作)通过 Stream Groupings 进行连接的

图。Storm 模型对复杂事件处理进行威胁的协作式检测的流程如图 8 所示。

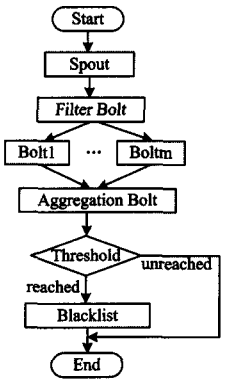


图 8 Storm 模型数据流图

Storm 模型的特点如下：

1)分布式；

2)并行计算；

3)数据一直在内存中流转，进程常驻内存；

4)容错性好，能管理工作进程和节点的故障；

5)计算逻辑部分的代码需要自实现，支持各种编程语言，默认支持 Clojure，Java，Ruby 和 Python；

6)对于复杂运算，运算模型支持 DAG(有向无环图)；

7)每个计算单元之间的数据通过 ZeroMQ 高效地传递，不持久化数据；

8)在一个小集群中，每秒可以处理数以百万计的消息。

Storm 模型的优势是可靠的消息处理，保证每个消息都会得到处理，当任务失败时，它负责从消息源重试消息。Storm 模型的运算结果可直接反馈到最终结果集中(展示页面、数据库、搜索引擎的索引)，因此，常用在实时处理、实时监控的系统中。

### 2.5 分布式多迭代批处理模型 Spark

Spark 模型[9]是一种分布式的多迭代批处理模型，由 Master 和 Workers 组成。用户在 Master 中定义对弹性分布式数据集(Resilient Distributed Datasets，RDD)的转换与操作，并把相应的操作信息发送到 Workers 节点。Workers 存储着数据分块并享有集群内存，是运行在工作节点上的守护进程，它收到对 RDD 的操作时，根据数据分片信息进行本地化数据操作，生成新的数据分片、返回结果或把 RDD 写入存储系统。Spark 模型如图 9 所示。

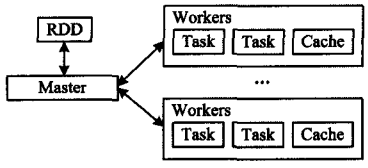


图 9 Spark 模型

Spark 模型对数据流的处理是通过将用户定义的一系列 RDD 转化成 DAG 图，DAG Scheduler 把 DAG 转化成一个 TaskSet，TaskSet 向集群申请计算资源，集群把 TaskSet 部署到 Worker 中进行运算。Spark 模型对用户定义的 RDD 进行处理的数据流程如图 10 所示。

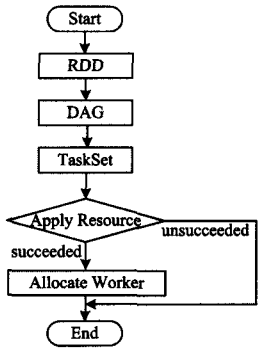


图 10 Spark 模型数据流图

Spark 模型的特点如下：

1)分布式；

2)超快速数据处理，使用 RDD 使得它可在内存中透明地存储数据并在需要时才传输到磁盘，减少对磁盘的读写次数；

3)支持复杂查询，如 Map 和 Reduce 操作、SQL 查询、流数据，还可优化迭代工作负载；

4)实时流处理，使用 Spark Streaming 来操纵实时数据；

5)计算逻辑部分的代码需要自实现，支持各种编程语言，如 Scala，Java 和 Python 等；

6)运行速度在内存中是 Hadoop MapReduce 的 100 倍，在磁盘中是 Hadoop MapReduce 的 10 倍。

Spark 模型可以独立运行，也可以与 Hadoop 集成，从任何 Hadoop 数据源读取数据，如 HBase，HDFS 等。

因此，Spark 是一个高效的分布式计算模型，比较适用于离线、快速的大数据处理。

### 2.6 协作式检测模型比较

通过分析 5 种典型模型可以得出：Esper 适合集中式地部署在小规模实时处理的网络环境中；Hadoop 适合分布式地部署在大规模批处理的网络环境中；Agilis 适合分布式地部署在大规模实时处理的网络环境中；Storm 适合分布式地部署在大规模容错实时处理的网络环境中；Spark 适合分布式地部署在大规模离线快速处理的网络环境中。这 5 种模型的比较如表 1 所列。

表 1 协作式检测模型比较

| Model | Deployment | Process | Language | Logic | Open Source |
|---|---|---|---|---|---|
| Esper | centralized | real-time | Java | EPL | yes |
| Hadoop | distributed | batch | Java | self-programming | yes |
| Agilis | distributed | real-time | Java | JAQL | no |
| Storm | distributed | fault-tolerant, real-time | Clojure | self-programming | yes |
| Spark | distributed | iteration batch | Scala | self-programming | yes |

## 3 协作式检测模型相关应用

在开放式的网络环境中，攻击者常用的攻击手段主要有分布式拒绝服务攻击(Distributed Denial of Service，DDoS)、中间人攻击(Man-in-the-Middle Attack，MITM 攻击)、高级持续性威胁(Advanced Persistent Threats，APT)等。

### 3.1 DDoS 攻击

DDoS 攻击指在不同位置的多个攻击者同时向一个或几

个目标发起攻击，用很多受控计算机向目标发起协作性的DOS攻击。DDoS攻击原理如图11所示。

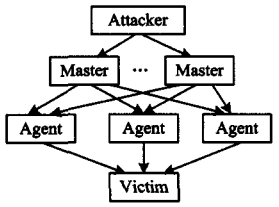


图11 DDoS攻击原理

由图11可知，一个比较完善的DDoS攻击包括以下4种角色。

1)攻击者：黑客所使用的机器，它是攻击主控台，控制整个攻击过程并向主控端发送攻击命令。

2)主控端：攻击者非法侵入并控制的一些主机，攻击者在这些主控端上安装特定的程序，以便使它可以接受攻击者发来的特殊命令，并能够把这些命令发送到代理攻击端主机上。

3)代理攻击端：也是攻击者侵入并控制的一批主机，是攻击的执行者，它们接收和运行主控端发来的命令并运行攻击程序，向受害者主机发送攻击。

4)受害者：被攻击的目标主机或服务器。

DDoS攻击采用C/S模式，主控端程序能在几秒钟内激活成百上千次代理攻击端程序，此类攻击是网络安全的最大威胁之一。

DDoS攻击有很多攻击工具。Trinoo[11]是一个消耗带宽的攻击工具，用一个或多个IP地址发起协作性的UDP洪泛攻击；Sharft[12]是Trinoo的一个派生工具，可以独立运行，也可以联合UDP/TCP/ICMP洪泛攻击发起攻击；TFN（Tribe Flood Network）[13]给攻击者提供可以同时发起损耗带宽和资源的攻击；TFN2K[14]增加了对构成攻击的所有组成部分之间的通信消息进行加密；Mstream[15]是一个点对点TCP ACK洪泛攻击工具，通过改变TCP数据包的ACK标志来攻击目标。

很多DDoS攻击数据来自MIT DARPA Intrusion detection project。它采用Mstream DDOS工具，通过TCP和UDP数据包进行传输，但是通信过程没有加密；主攻击者通过远程登录到受害者的机器，但是访问操作是经过密码保护的，这个特征在其它的DDoS攻击工具中是不存在的。

当一个提供用户接口和控制服务器的软件Mstream主控端被安装到一个受害的主机上，一个真正产生并发送DDOS攻击包的软件Mstream服务端被安装到几个受害的主机上时，DDoS攻击通过5个阶段展开攻击：

1)从远程站点进行IP扫描；

2)探查活跃的IP，寻求主机上运行的具有漏洞的某个守护进程；

3)通过守护进程的脆弱点成功或不成功地闯入到这些主机上；

4)在这些主机上安装Trojan Mstream DDoS软件；

5)发起DDoS攻击。

DDoS攻击的第一步是对主机端口进行扫描，检测出远程或本地主机含有脆弱点的程序，为下一步继续实施攻击作准备。这也是很多攻击常用的方法，因此，对主机端口扫描的检测为下一步将攻击完全检测出来作准备。

Esper模型利用10台虚拟机搭建一个测试床，在一台虚拟机上部署Esper复杂事件处理引擎，剩余的9台虚拟机通过模拟9个入侵者来提供入侵踪迹数据，并且还利用一个开源的仿真器WANem[16]模拟了一个大规模部署环境以使所有虚拟机能够相互连接。入侵踪迹数据是从ITOC研究网站[17]、LBNL/ICSI企业追踪项目[18]、MIT DARPA入侵检测项目[19]下载得到的，这些踪迹数据被拆分成9个子踪迹来模拟9个入侵者的行为。通过大量的实验得出，DDoS攻击中的端口扫描可以被Esper模型检测出来[20]。

Agilis模型利用7台虚拟机搭建一个测试床，在一台虚拟机上部署Agilis管理组件，即Job Tracker和WXS目录服务器，剩余的6台虚拟机部署Task Tracker和WXS容器，并通过模拟6个入侵者来提供入侵踪迹数据，同样还利用一个开源仿真器WANem模拟了一个大规模部署环境以使所有虚拟机能够相互连接。入侵踪迹数据是从ITOC研究网站[17]和MIT DARPA入侵检测项目[19]下载得到的，这些踪迹数据被拆分成6个子踪迹来模拟6个入侵者的行为。通过大量的实验得出，DDoS攻击中的端口扫描可以被Agilis模型检测出来[21]。

Storm模型的检测方法与Esper模型的检测方法相同，大量的实验结果表明，DDoS攻击中的端口扫描可以被Storm模型检测出来[22]。

### 3.2 MITM攻击

MITM攻击是通过劫持或拦截合法用户和服务器之间的服务来实现的，服务被劫持之后，攻击者就可以转发客户的认证证书给服务器，然后充当客户与服务器之间后续事务的中间人。

典型的MITM攻击手段有会话劫持、DNS欺骗等技术，而常见的会话劫持包括TCP会话劫持、HTTP会话劫持和SSL会话劫持等。

假设主机C和主机S进行一次TCP会话，M为攻击者。

(1)TCP会话劫持过程如下：

1)C->S

2)S->C

3)C->S

4)S->C

5)C(攻击者M冒充的)->S

6)S->C

这样，主机S执行了攻击者M冒充主机C发送过来的命令，并且返回给主机C一个数据包；但是，主机C并不能识别主机S发送过来的数据包，所以主机C会以期望的序列号返回给主机S一个数据包，随即形成ACK风暴；如果解决了ACK风暴(例如ARP欺骗)，就可以进行TCP会话劫持。

(2)HTTP会话劫持过程如下：

1)从C到S的TCP会话正常建立；

2)C发出的HTTP搜索请求被一个离C很近的设备劫持，伪造来自S的数据包，并且很快返回HTTP回复；

3)来自伪造S的数据包继续完成与C的TCP会话过程，之后该TCP会话被来自伪造S的数据包正常终止；

4)此时，真正来自S的数据包才到达C，故C向S发送连接不可用。

(3)SSL会话劫持(主机M通过数据流重定向技术，使得

主机 C 与主机 S 之间的通信流量都流向主机 M)过程如下：

1)主机 C 与主机 S 建立 SSL 连接，但发送的连接建立请求被重定向到主机 M；

2)主机 C 与主机 M 建立 TCP 连接，然后向主机 M 发起 SSL 连接请求；

3)主机 M 收到主机 C 的连接请求后，首先与主机 S 建立 TCP 连接，然后向主机 S 发起 SSL 连接请求；

4)主机 S 响应主机 M 的请求，主机 M 与主机 S 之间成功建立 SSL 连接；

5)主机 M 伪造数据，并响应来自主机 C 的 SSL 连接请求；

6)主机 C 与主机 M 成功建立 SSL 连接；

7)之后，对于主机 C 发往 SSL 服务端的数据，主机 M 可以捕获并解密查看；

8)对于主机 S 返回给 SSL 客户端的数据，主机 M 也可以捕获并解密查看。

至此，主机 M 实现了完整的 SSL 中间人的会话劫持。

(4)DNS 欺骗过程如下：

1)主机 C 向 DNS 服务器 S 发送请求域名对应的 IP 地址；

2)主机 C 却将 DNS 请求发送到攻击者 M；

3)攻击者 M 伪造 DNS 响应，将正确的 IP 地址替换为其他含有恶意网页的 IP 地址。

这样，主机 C 访问的是攻击者 M 提供的恶意网页，而不是主机 C 想要取得的网站主页。

随着无线局域网技术的发展，在 Wi-Fi 环境中进行 MITM 攻击也是非常容易的，Wi-Fi 中间人攻击过程如下：

1)攻击者使用与合法接入点相同的 SSID(若加密，认证/加密算法也须相同，并且预共享秘钥相同)；

2)相对被攻击者而言，让伪造接入点的功率大于合法接入点；

3)攻击者以客户端的身份连接到合法接入点，在中间中转被攻击者与合法接入点之间的流量。

这样，客户与合法接入点之间的数据都能够被明文监听，攻击者可以进一步实施更高级别的攻击手段。

可见，在以获取经济利益为目标的黑客技术越来越多的情况下，MITM 攻击成为对网银、网游、电子支付、网上交易等最具威胁、最具破坏性的一种攻击手段。而将 Esper 模型与 Agilis 模型相结合，建立一个覆盖网络便可检测出 MITM 攻击。

事件源是由 3 个金融机构提供的，它们都向 Agilis 的处理容器 Esper 引擎提供客户每一次登录到某个金融机构站点时产生的事件。每条事件包含的信息用一个五元组表示：(客户登录时的 IP 地址，客户所登录的服务的标识符，客户的标识符，生成该事件的金融机构的标识符，客户登录的时间)。为了确保匿名性及隐私性，这些标识符不一定需要提供真实值，可以用别名对其进行标识。

MITM 攻击的一个特征可能是在一个短的时间窗口内来自同一个 IP 地址的多个不同客户向同一个服务进行了多次成功访问。因此，可根据事件中源 IP 地址的标识符、服务的标识符生成消息 ID。可见，来自同一个 IP 且访问同一个服务的事件将被转发给同一个 Esper 引擎进行处理，由此判断在一个时间窗口内的访问次数是否达到门限值。当在一个特定的时间窗口内具有同样标识符的事件达到或超过门限值时，Esper 引擎预定义的模式被确认并发出报警；如果这些事件的 IP 地址之前未被怀疑过，则时间窗口不变；如果其 IP 之前已被怀疑过，则再自动增加时间窗口。因此，MITM 起初是不能被检测的，但是一旦收到 Agilis 模型端口扫描的报警信息，便自动增加可疑 IP 地址的时间窗口，这种不需人为参与的自主反应可检测出部分 MITM 攻击，从而在一定程度上降低安全风险。

### 3.3 APT 攻击

APT 攻击是利用先进的攻击手段对特定目标进行长期、持续性网络攻击的攻击手段，攻击的原理相对于其他攻击手段更为高级和先进。主要体现在 APT 在发动攻击之前需要对攻击对象的业务流程和目标系统进行精确的收集，在收集的过程中，此攻击会主动挖掘被攻击对象受信系统和应用程序的漏洞，利用这些漏洞组建攻击者所需的网络并利用 0day 漏洞进行攻击。

APT 攻击的主要特征如下。

1)潜伏性：攻击和威胁可能在用户环境中持续一年以上或更久，不断收集各种信息直到收集到重要情报；

2)持续性：不断尝试各种攻击手段；

3)锁定特定目标：针对被锁定对象寄送社交工程恶意邮件，从而在计算机中植入恶意软件；

4)安装远程控制工具：建立类似僵尸网络 Botnet 的远程控制架构，定期传送有价值文件的副本给命令和控制服务器(C&C Server)审查，将过滤后的敏感机密数据利用加密的方式外传。

APT 攻击具有检测难、持续时间长和攻击目标明确等特征。此种攻击的特点在于隐匿自己，针对特定对象长期地、有计划性地、有组织性地窃取数据。入侵客户的途径主要有：1)智能手机、平板电脑、USB 等移动设备；2)社交工程的恶意邮件；3)防火墙、服务器等系统漏洞。

通过上述途径成功实施 APT 攻击，需要以下 6 个步骤：情报搜集；首次突破防线；幕后操纵通讯；横向移动；资产/资料发掘；资料外传。总之，APT 攻击通过一切方式绕过传统的安全方案(如防病毒软件、防火墙、IPS 等)，长时间地潜伏在系统中，使得传统防御体系难以检测。从 2010 年起，在世界各国接连发生针对大型企业和国家重大项目的比较典型的 APT 攻击事件如图 12 所示。

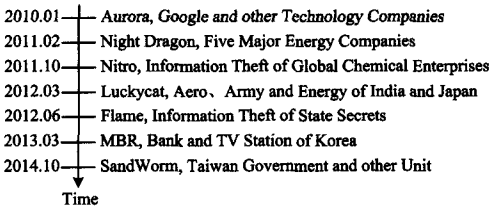


图 12 APT 攻击历史

APT 攻击是企业信息安全的一大隐忧，此类攻击的防御必须融入到一个更大的监测及预防策略中，因此防御 APT 攻击的基本措施是分布式多点部署、集中管控。在网络设备、终端、服务器上部署探针，采集网络设备的原始流量、终端和服务器的日志；对采集到的数据进行集中的海量数据存储和全面深入分析，还原 APT 攻击场景。由于 APT 攻击具有持久

性，且采集的数据需要持久化，不适合采用实时的模型对其进行检测，因此可以采用离线的模型（如 Hadoop 或 Spark），经过长期的数据存储和分析来发现 APT 攻击。

### 3.4 协作式检测模型应用比较

对于攻击者常用的攻击手段，可以单独使用一个协作式检测模型对其进行检测，也可使用几个协作式检测模型的组合对其进行检测。具体来讲，实施 DDoS 攻击的一个关键步骤是端口扫描，可以采用 Esper，Agilis 和 Storm 模型对其检测；对于 MITM 攻击，可以采用 Esper 模型与 Agilis 模型相结合的方式对其进行检测；对于 APT 攻击，可以采用 Hadoop、Spark 模型对其进行检测。

## 4 协作式检测架构模型部署

协作式检测架构模型包括发送和接收处理两个组件。发送数据的探针组件包括两个模块：1）发送网络数据的网络探针模块，既可以使用开源的软件获取网络数据，也可以自编程实现网络数据的获取；2）发送主机业务日志数据的主机探针模块，可以自编程实现主机业务日志数据的获取。接收处理组件主要是由对探针发送的复杂事件进行接收并分析处理的模型组成，这些模型可根据需要进行不同的架构部署。

### 4.1 探针部署

为了对威胁进行协作式检测，需要获取网络数据和业务日志数据，这些数据可通过部署网络探针和主机探针来获取。网络探针在网关、路由器、交换机等网络设备中部署 Tcpdump、Sniffer、Wireshark 等嗅探工具，实时监控并抓取网络中的数据；主机探针在终端、服务器上部署，实时获取业务的日志数据。同时，在网络的关键部位部署复杂事件处理引擎，对抓取的网络数据和业务日志数据进行分析和预处理，并对数据进行关联，发掘蓄意的攻击。

利用嗅探工具获得的网络数据常以两种形式存在，一种是以 dump 格式文件存储的网络数据，另一种是以 XML 格式文件存储的网络数据。

dump 格式文件是标准的二进制文件，一条网络数据包含 Tcpdump 格式文件头、Packet 头、IP 数据报包 3 部分。其中，Tcpdump 格式文件头的长度为 24 个字节，Packet 头的长度为 16 个字节，IP 数据报包长度由 IP 数据报首部的总长度字段决定。

dump 格式文件中的一条报警记录如下。

1）Tcpdump 格式文件头部分包含 4 字节的 Tcpdump 格式文件标记、2 字节的主版本号、2 字节的子版本号、4 字节的时区、4 字节的精确时间戳、4 字节的数据包大小、4 字节的数据链类型，共 24 字节，内容如下：

A1 B2 C3 D4 00 02 00 04 00 00 00 00 00 00 00 00 00 00 01 01 D0 00 00 00 01

2）Packet 头部分包含 8 字节的时间戳、4 字节的本次保存的 IP 数据报长度、4 字节的 IP 数据报原有长度，共 16 字节，内容如下：

55 2F 6D 1B 00 00 00 00 00 00 00 4A 00 00 00 4A

3）长度为 60 字节的 IP 数据报包内容为：

45 00 00 3C 8A 9B 40 00 40 06 23 D9 CA 4D A2 D5 AC 10 73 14 88 48 00 17 F9 84 64 6C 00 00 00 00 A0 02 80 00 C4 0F 00 00 02 04 05 B4 01 03 03 00 01 01 08 0A 00 05 94 5A 00 00 00 00

XML 格式文件中的一条报警记录如下。

```
〈Alert version="1" impact="unknown" alertid="1"〉
  〈Time〉
    〈date〉04/16/2015〈/date〉
    〈time〉16:04:43〈/time〉
    〈sessionduration〉00:08:08〈/sessionduration〉
  〈/Time〉
  〈Analyzer ident="tcpdump_inside"〉
    〈name〉tcpdump_inside〈/name〉
  〈/Analyzer〉
  〈Source spoofed="unknown"〉
    〈Node〉
      〈Address category="ipv4-addr"〉
        〈address〉202.77.162.213〈/address〉
      〈/Address〉
    〈/Node〉
  〈/Source〉
  〈Target〉
    〈Node〉
      〈Address category="ipv4-addr"〉
        〈address〉172.16.115.20〈/address〉
      〈/Address〉
    〈/Node〉
    〈Service〉
      〈name〉tcp〈/name〉
      〈sport〉34888〈/sport〉
      〈dport〉23〈/dport〉
    〈/Service〉
  〈/Target〉
〈/Alert〉
```

业务日志数据采用一个七元组（源 IP 地址、目的 IP 地址、开始时间、结束时间、HTTP 操作、URL、用户会话 ID）表示，其中一条日志数据的内容如下：204.97.171.187 1.1.1.1 2015-04-17 11:42:50 2015-04-17 11:43:56 GET/test/pay_bill.php HTTP/1.1 http://1.1.1.1/test/pay_bill.php 7e18371db830b37c30e1edd4c5560fbd。

网络探针和主机探针将获取的这些网络数据和业务日志数据发送给网络中部署的复杂事件处理引擎，引擎收到数据后对其进行预处理、关联、在线分析、存储、离线分析，从而协作式地检测网络和业务中存在的威胁或攻击。

### 4.2 架构模型部署

协作式威胁检测模型有集中式、分布式、在线型、离线型这 4 种类型，这些检测模型对事件进行分析处理的规则依赖于处理容器所实现的算法，与协作式威胁检测架构本身无关，因此可根据需要部署相应的模型或将几个模型进行组合形成新的模型。下面给出两种应用网络架构模型的典型部署方案。

1）对等网络：对于需要实时处理的对等网络，由于 Esper 会出现漏报的情况，协作式威胁检测架构可分别部署 Storm 和 Agilis 模型或两个模型的任意组合；对于需要离线处理的对等网络，协作式威胁检测架构可分别部署 Hadoop 和 Spark 模型或两个模型的组合。

2)分等级的安全域网络:对于实时处理的分等级安全域网络,在低级别安全域,Esper 漏报不会产生太大的影响,协作式威胁检测架构可分别部署 Esper,Storm,Agilis 模型或 3 个模型的任意组合;在高级别安全域,Esper 漏报会产生决定性作用,协作式威胁检测架构可分别部署 Storm 和 Agilis 模型或两个模型的任意组合;对于需要离线处理的分级安全域网络,协作式威胁检测架构可分别部署 Hadoop 和 Spark 模型或两个模型的组合。因此,对于分等级的安全域网络,协作式威胁检测架构模型需要根据安全域级别分级部署。在两个等级的安全域网络中进行威胁的协作式实时检测时,Esper 与 Storm 组合的模型如图 13 所示。

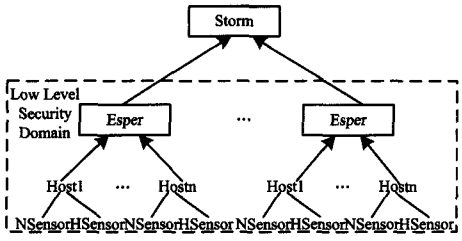


图 13 Esper 与 Storm 组合的实时检测模型

由图 13 可知,在分等级的安全域网络中部署威胁的协作式检测架构模型时,下层安全域模型实时处理并将汇聚结果传输给上层安全域模型进行实时处理,随着安全域等级的升高,可以一直向上延伸部署,从而构成了一个层次结构。因此,在分层结构的网络中部署威胁的协作式检测架构模型类似于分等级的安全域网络中的布置情况。特别地,在较大的机构中,其组建的网络常常是跨城区的,因此在分机构中部署威胁的协作式实时检测架构模型类似于在低级别安全域中的部署情况;而在总部机构中部署威胁的协作式实时检测架构模型则类似于在高级别安全域中的部署情况。

**结束语** 恶意的不法分子采用直接或间接的方法攻击个人、机构、国家,从而使其受到不同程度的威胁。由于此类信息与正常的业务活动同时在网络中传输,使得数据的形式多种多样,数据量巨大,而且需要被高速地处理,其具备大数据的特点。因此,为了在大数据环境中检测出存在的威胁,本文首先对协作式检测模型进行分析、比较,阐述不同模型所适用的网络环境;其次,对网络环境中常用的攻击手段进行分析,说明检测这些攻击适合采用的模型;然后,指出在大数据环境下对威胁的协作式检测架构模型的部署方案,包括发送数据的探针组件和接收并分析处理探针所发送的复杂事件的模型,这些模型可根据实际需要进行不同的架构部署,特别地,给出了对等网络、分等级的安全域网络、分层结构网络中架构模型的部署方案。

## 参 考 文 献


[1] Global Fraud Report-Annual Edition 2011-2012, Kroll [EB/OL]. http://www.krollconsulting.com/fraud-report/2011-12/press-only

[2] In the Crossfire:Critical Infrastructure in the Age of Cyber War [J/OL]. http://www.mcafee.com/us/resources/reports/rp-in-crossfire-critical-infrastructure-cyber-war.pdf

[3] Giuseppe Antonio Di Luna. A Collaborative Processing System for Cyber Attacks Detection and Crime Monitoring [D]. Rome: Sapienza University,2010

[4] EsperTech: Event Series Intelligence [EB/OL]. http://www.espertech.com

[5] Apache Software Foundation. Welcome to ApacheTM Hadoop? [EB/OL]. http://hadoop.apache.org

[6] Aniello L,Baldoni R,Chockler G,et al. Agilis:An Internet-Scale Distributed Event Processing System for Collaborative Detection of Cyber Attacks [R]. MIDLAB Technical Report,2011

[7] Storm. Distributed and fault-tolerant realtime computation [EB/OL]. http://storm-project.net

[8] Spark. Lightning-fast cluster computing [EB/OL]. http://spark.apache.org

[9] Beyer K,Ercegovac V,Gemulla R,et al. JAQL:A scripting language for large scale semistructured data analysis [J]. Proceedings of the VLDB Endowment,2011,4(12):1272-1283

[10] Hunt P,Konar M,Junqueira F P,et al. Zookeeper: Wait-free coordination for internet-scale systems [C]// Usenix Annual Technical Conference. Berkeley,CA:Usenix,2010

[11] Dittrich D. The DoS Project's "trinoo" distributed denial of service attack tool [EB/OL]. https://staff.washington.edu/dittrich/misc/trinoo.analysis

[12] Dietrich S,Long N,Dittrich D. Analyzing Distributed Denial of Service tools:the Shaft Case [C]//Proceedings of the 14th Systems Administration Conference (LISA 2000). New Orleans,LA,USA,2000:329-339

[13] Dittrich D. The Tribe Flood Network Distributed Denial of Service attack tool [EB/OL]. https://staff.washington.edu/dittrich/misc/tfn.analysis

[14] Barlow J. TFN2K-an analysis [EB/OL]. http://packetstormsecurity.com/distributed/TFN2k_Analysis-1.3.txt

[15] Dittrich D,Weaver G,Dietrich S,et al. The _mstream_ Distributed Denial of Service attack tool [EB/OL]. ttps://staff.washington.edu/dittrich/misc/mstream.analysis.txt

[16] WANem-Wide Area Network Emulator [EB/OL]. http://sourceforge.net/projects/wanem/files/WANem/

[17] ITOC research:CDX datasets [OL]. http://www.itoc.usma.edu/research/dataset/index.html

[18] LBNL/ICSI enterprise tracing project[OL]. http://www.icir.org/enterprise-tracing/download.html

[19] 2000 DARPA intrusion detection scenario specific data sets [OL]. http://www.ll.mit.edu/ideval/data/2000data.html

[20] Aniello L,Luna G A D,Lodi G,et al. Collaborative Inter-domain Stealthy Port Scan Detection Using Esper Complex Event Processing [C]//Roberto Baldoni,Gregory Chockler. Collaborative Financial Infrastructure Protection. Springer,2012:139-156

[21] Aniello L,Baldoni R,Chockler G,et al. Distributed Attack Detection Using Agilis [C]//Roberto Baldoni,Gregory Chockler. Collaborative Financial Infrastructure Protection. Springer,2012:157-174

[22] Lodi G,Aniello L,Luna G A D,et al. An event-based platform for collaborative threats detection and monitoring [J]. Information Systems,2014,39:175-195