计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20200902
58(5): 995-1005, 2021

一种无监督的窃密攻击及时发现方法

冯 云 刘宝旭 张金莉 汪旭童 刘潮歌 申明喆 刘奇旭
(中国科学院信息工程研究所 北京 100093)
(中国科学院大学网络空间安全学院 北京 100049)
(fengyun@iie.ac.cn)

An Unsupervised Method for Timely Exfiltration Attack Discovery

Feng Yun, Liu Baoxu, Zhang Jinli, Wang Xutong, Liu Chaoge, Shen Mingzhe, and Liu Qixu
(*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)
(*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049)

Abstract In recent years, exfiltration attacks have become one of the severest threats to cyber security. In addition to malware, human beings, especially insiders, can also become the executor of the attack. The obvious anomalous digital footprint left by an insider can be minuscule, which brings challenges to timely attack discovery and malicious operation analysis and reconstruction in real-world scenarios. To address the challenge, a method is proposed, which treats each user as an independent subject and detects the anomaly by comparing the deviation between current behavior and the normal historical behavior. We take one session as a unit to achieve timely attack discovery. We use unsupervised algorithms to avoid the need for a large number of labeled data, which is more practical to real-world scenarios. For the anomalous session detected by the algorithm, we further propose to construct event chains. On the one hand, it can restore the specific exfiltration operation; on the other hand, it can determine the attack more accurately by matching it with the exfiltration attack mode. Then, the experiments are undertaken using the public CMU CERT insider threat dataset, and the results show that the accuracy rates were more than 99%, and there were no false-negative and low false-positive, demonstrate that our method is effective and superior.

Key words exfiltration attack discovery; user events; insider threat detection; unsupervised algorithm; clustering; event chain

摘 要 近年来,窃密攻击成为了最严重的网络安全威胁之一.除了恶意软件,人也可以成为窃密攻击的实施主体,尤其是组织或企业的内部人员.由人实施的窃密很少留下明显的异常痕迹,给真实场景中攻击的及时发现和窃密操作的分析还原带来了挑战.提出了一个方法,将每个用户视为独立的主体,通过对比用户当前行为事件与其历史正常行为的偏差检测异常,以会话为单元的检测实现了攻击发现的及时性,采用无监督算法避免了对大量带标签数据的依赖,更能适用于真实场景.对算法检测为异常的会话,进一步提出事件链构建方法,一方面还原具体窃密操作,另一方面通过与窃密攻击模式对比,更精确

收稿日期:2020-11-07;**修回日期**:2021-02-23
基金项目:国家自然科学基金项目(61902396);中国科学院青年创新促进会(2019163);中国科学院战略性先导科技专项项目(XDC02040100);中国科学院网络测评技术重点实验室资助;网络安全防护技术北京市重点实验室资助
This work was supported by the National Natural Science Foundation of China (61902396), the Youth Innovation Promotion Association of Chinese Academy of Sciences (2019163), the Strategic Priority Research Program of Chinese Academy of Sciences (XDC02040100), the Project of the Key Laboratory of Network Assessment Technology at Chinese Academy of Sciences, and the Project of Beijing Key Laboratory of Network Security and Protection Technology.
通信作者:张金莉(zhangjinli@iie.ac.cn)

地判断攻击.在卡内基梅隆大学的 CERT 内部威胁数据集上进行了实验,结果达到 99%以上的准确率,且可以做到无漏报、低误报,证明了方法的有效性和优越性.

关键词 窃密攻击发现;用户事件;内部威胁检测;无监督算法;聚类;事件链

中图法分类号 TP391

窃密攻击是企业和组织都正在面临的严峻问题,威胁着知识产权和机密数据的安全.窃密攻击可以由外部攻击者投递的恶意软件实施,也可以由内部人员实施,即内部威胁.根据 Verizon 统计的数据[1],大约 30%的数据泄露是由内部人员造成的.Ponemon 研究所 2020 发布的《内部威胁成本全球报告》[2]也显示,由内部威胁造成的数据泄露成本在 2 年间增长了 31%,达到 1 000 余万美元.内部人员可能出于受贿、将要离职等原因窃取公司数据,由于内部人员更加熟悉企业内部的部门架构、内网环境和安全策略,并且拥有一定的内部计算机访问权限,因此带来的威胁更大.

在攻击检测和发现领域已经有很多研究成果.入侵检测和恶意软件检测是典型的方法,但是在内部人员窃密攻击检测方面存在缺陷.通常,在窃密攻击尤其是由人实施的窃密攻击中,很少有明显的异常行为,而是由一系列正常的操作构成,如文件、网络和电子邮件操作等,并且很少使用恶意软件,因此不会被入侵检测系统触发警报,传统的检测方法难以生效.另外,存在一些策略上的解决方案,例如安全控制策略、数据防泄露等,但仅可以防范未授权的访问.相反,内部人员窃密是一系列的用户行为事件(后文简称"用户事件"),因此,利用用户事件进行攻击发现值得研究.

现有的内部威胁检测方法更关注恶意内部人员检测,而不是攻击发现.很多工作从用户长期完整的事件日志中提取特征,然后通过基于统计学或机器学习的方法,将正常用户和恶意用户进行分类,从而实现恶意人员检测.但这类方法需要大量带标签的数据作为训练集,且会造成攻击发生和攻击发现之间的时间滞后性.另一种内部威胁检测方法是基于规则的,依赖于专家知识,且需要人工更新规则来避免攻击者刻意的伪装和策略转换.目前,很多研究试图采用深度学习技术以实现更智能的内部人员攻击检测,但这也需要大量的数据,且会以攻击发现的及时性作为代价.

本文设计并实现了一个窃密攻击及时发现方法.我们关注用户事件从而解决由人实施的攻击,尤其是内部人员威胁.为了攻击发现的及时性,以会话为单位进行用户事件日志的分析,会话即用户从登入计算机到登出计算机的阶段.另外,由于在真实的场景中很难获得大量的带标签数据,本文利用无监督算法进行异常检测.同时,我们针对的是攻击发现而不是恶意内部人员检测.更具体地说,发现什么人在什么时候做了什么,而不只发现是什么人做的.本文主要贡献有 4 个方面:

1) 提出利用用户事件发现攻击行为,基于用户行为习惯构建行为模式,以会话为单位,通过比较用户当前行为与历史行为的差异实现异常检测,并加入了一个动态更新机制以适应用户可能的习惯变化;

2) 对于无法与正常行为模式匹配的异常事件序列,利用无监督算法进行检测,对每一个当前的事件序列进行处理以实现攻击发现的及时性;

3) 对于算法检测为异常的会话,进行事件链的构建以还原具体的恶意窃密操作,并进行更精确的攻击判断;

4) 进行了实验比较不同算法的效果,同时证明了方法的有效性.

1 相关工作

很多研究人员提出了针对不同场景的攻击检测和发现方法.随着攻击规模和复杂度的提高,所带来的挑战也越来越严峻.尤其是对于窃密攻击,因其很少有明显的异常行为,更加难以发现.

事件起源追踪法是目前的研究热点,通过监控并分析系统中的所有实体及事件来检测攻击,并构建攻击事件链.有些研究聚焦事件起源关系的捕获和记录,通过改进系统级进程启动和中断追踪机制实现更全面、高效的事件起源信息捕获[3-6].利用细粒度的起源信息进行事件间因果关系分析从而检测攻击也是一个研究重点.文献[7]提出利用进程、文件等实体间的交互信息,在起源图上随机游走并计算异常分数,从而发现可疑的事件路径.文献[8]设计了 3 种属性来描述一个事件的优先级,包括稀有度、扇出度和数据流终止,以进行及时的异常事件

链构建.由于系统中事件数量庞大,起源图可能会非常错综复杂,称为依赖关系爆炸问题.文献[9]提出了 HOLMES,通过评估依赖关系的强弱来减少不重要的事件.另一个方法 UNICORN[10]构建了一个可增量更新、固定大小的图形数据结构,以跟踪系统的整个起源历史并检测攻击.然而,这些系统都关注进程级的事件.

内部威胁和商业间谍是窃密攻击的常见类型,这 2 种攻击都是由人而不是程序执行的.因此,需要关注利用用户事件的攻击发现.传统的方法是从用户行为日志中提取特征,利用统计方法、机器学习或深度学习算法进行异常分析.一般来说,特征包括基于频率、基于时间、基于序列和基于属性等[11-14].由于用户事件序列可以看作一个事件链,因此隐马尔可夫模型可以用于攻击检测[15-16].文献[17]采用了多种机器学习算法,包括逻辑回归、随机森林和神经网络等来识别内部威胁.Jiang 等人[18]提出了一种将用户之间的特征和属性刻画成图的图卷积网络(graph convolutional network, GCN)模型.文献[19]将特征转化为灰度图像,并通过预先训练的深度卷积神经网络进行图像分类,检测恶意用户.近年来,为了提高方法的自动化程度,针对特征自动学习出现了一些研究成果.利用自然语言处理(natural language processing, NLP)算法 Word2Vec 将行为日志转换为向量,通过向量之间的距离来度量事件之间的依赖关系是近年来备受关注的方法[20-22].Liu 等人[23-24]提出了 4W('who','when','where','what')句子模板将每个事件转换成文本,然后使用 Word2Vec 将每个事件转换成向量.另一种方法是使用长短期记忆网络(long short-term memory, LSTM)学习用户行为并提取特征,将这些特征作为卷积神经网络(convolutional neural network, CNN)分类器的输入[25].然而,这些方法依赖于一定数量的数据或标签,不能很好地适用于真实的内网场景和及时的攻击发现需求.

本文针对现有研究存在的问题进行了改进.1)仅基于行为日志进行分析,保证了方法的通用性;2)利用当前而非长期累计的事件日志来发现攻击,保证了攻击发现的及时性;3)利用无监督算法进行检测,无需借助大量带标签数据训练模型,保证了方法的实际意义;4)能够构建可视的事件链以还原具体恶意操作,有助于攻击分析、溯源和取证.表 1 对比并展示了本文方法如何优于最新的其他研究工作.

Table 1　Comparison with Related Work

表 1　与相关研究工作的对比

文献方法	恶意人员识别	恶意会话识别	恶意事件识别	事件链构建	时效性	数据使用
文献[13]	√	×	×	×	基于每日或固定周期日志	行为日志及心理测量数据
文献[14]	√	×	√	×	及时发现	行为日志及邮件内容、邮件情感倾向、网页关键词等
文献[21]	√	×	√	×	基于 2 个月的日志	行为日志
文献[23]	√	×	√	×	基于 1 周的日志	行为日志及网络传输数据、邮件主题、元数据等
文献[25]	√	×	×	×	基于所有日志	行为日志
本文方法	√	√	√	√	及时发现	行为日志

注:"√"表示该方法支持;"×"表示该方法不支持.

2　本文方法设计

2.1　概　述

本文抛弃传统方法中不同用户间对比分类的策略,而是将每个用户视为独立的个体,从用户事件中观察其当前行为相比历史行为的异常变化.具体来说,每个用户都有独特而相对稳定的行为习惯,因此能够利用历史行为日志为每个用户建立正常行为模式.用户行为模式可以用来衡量当前的行为是否正常.本文以会话为单位进行用户事件日志的分析,而非几周甚至几个月的日志,以实现攻击发现的及时性.采用无监督算法检测用户当前行为的异常,然后对异常会话中的事件序列进行可视化的攻击事件链构建,还原用户具体的恶意操作,并进一步分析判断是否真正发生了窃密攻击.

方法的流程如图 1 所示,由 4 个组件构成:

1) 用户行为模式构建.用户行为模式基于用户在特定时间段内的良性、正常的日常行为事件而构建.

2) 实时监控.对用户事件进行持续的实时监控和分析.

3) 检测算法.采用无监督算法进行异常会话的检测.

4) 事件链构建.对检测为异常的会话进行事件链的构建,从而直观地掌握用户的恶意窃密操作,并最终进行更精确的攻击判断.

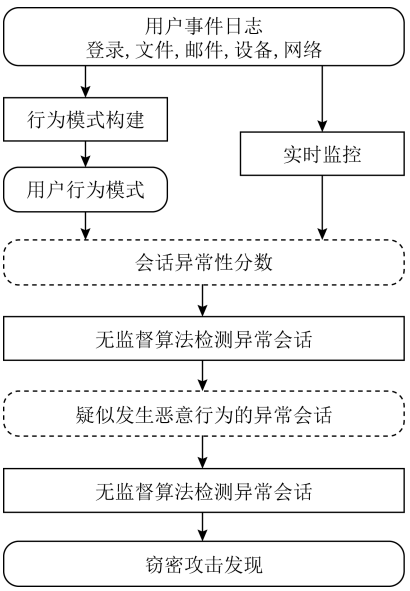

Fig. 1 Approach schematic overview of this paper

图 1 本文方法流程示意图

2.2 用户行为模式构建

用户通常具有独特且相对稳定的行为习惯.如果用户执行了一个罕见操作,偏离了他的正常行为模式,则可能是恶意操作.因此,我们选取一段特定的时间作为训练期,收集并分析用户在训练期内的良性、正常的行为事件,用于构建用户行为模式.关注 7 个类别的用户事件,包括登入事件、登出事件、文件操作事件、电子邮件发送事件、电子邮件接收事件、可移动设备连接事件和网络活动事件.在这里,登入、登出事件的时间特征不同,故区分对待;而电子邮件发送是窃密渗出的重要手段,且邮件发送、邮件接收的发件人、收件人含义不同,因此把电子邮件发送、接收事件进行区分.

首先,统计事件类稀有度,即每一类事件发生的稀有程度,取发生了该类别事件的会话数量所占全部会话数量的比例为值.对于这一特征,关注除登入、登出事件以外的 5 类事件,因为登入、登出事件在每个会话中都必然发生.其次,共设计了 33 个维度的指标以更细致地描述事件的特征,详细信息如表 2 所示.对于每一个指标,统计所有可能情况的发生频率以评估每种情况的稀有程度.例如,对于用户登入时间这一指标,统计他在 1 天 24 小时中每个小时登入的频率.事件类稀有度和每个维度指标稀有度共同构成了用户行为模式.基于用户行为模式,可以更快地筛选出稀有的事件,事件越稀有,则其恶意性可能越大.例如,某用户通常在早上 7 点登入自己的计算机,但某天在凌晨登入,这是一个稀有事件,意味着他可能执行一些不可见人的操作.

Table2 Indicators to Build User Profile

表 2 用于构建用户行为模式的指标

事件类别	指标维度
登入事件	在 1 天中的哪个小时登入 在 1 星期中的哪天登入 登入的计算机编号 登入的计算机是否属于自己
登出事件	在 1 天中的哪个小时登出 在 1 星期中的哪天登出 登出的计算机编号 登出的计算机是否属于自己
文件事件	文件类型 文件名 文件路径 文件操作类型 是否来自/移入可移动设备 操作时间
电子邮件发送事件	接收方的用户名 接收方的域名 发送方的用户名 发送方的域名 是否包含附件 操作类型 操作时间
电子邮件接收事件	接收方的用户名 接收方的域名 发送方的用户名 发送方的域名 是否包含附件 操作类型 操作时间
可移动设备连接事件	连接计算机编号 操作时间
网络事件	操作类型 访问的 URL 域名 操作时间

为了对异常性进行量化评估,本文设计了会话异常性分数计算方法.训练期分数的计算与实时行为监控阶段的分数计算方法基本相同,将在 2.3 节详细描述.

2.3 实时监控

会话异常性分数计算流程如图 2 所示:

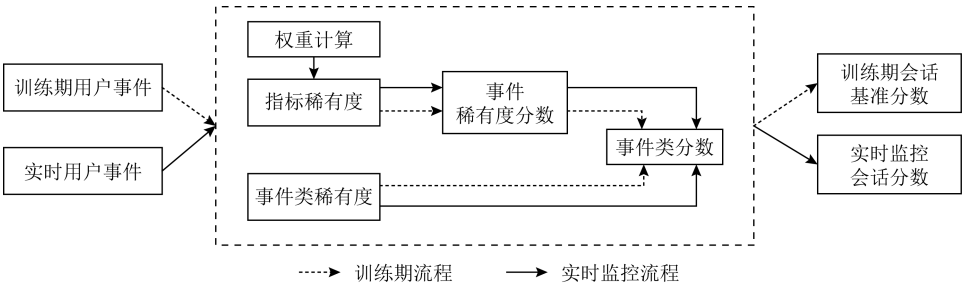

Fig. 2　Session anomaly score calculation process

图 2　会话异常性分数计算流程

持续监控当前用户事件,从而及时发现窃密攻击.关注用户在每个会话中的每个事件,并利用 2 方面稀有程度进行异常性评分,包括事件类稀有度 r_{etype}、各维度指标稀有度 $r_{indicator}$.

对于某一指标 $indicator_j$ 的某一种情况 x_i 的稀有度 $r_{indicator_j}(x_i)$ 计算:

$$r_{indicator_j}(x_i)=\begin{cases}0, \text{若 } x_i \text{ 未发生;}\\ \dfrac{N_{x_i}}{N_x}, \text{若 } x_i \text{ 发生,}\end{cases} \quad (1),$$

其中,N_{x_i} 表示 x_i 发生的次数,N_x 表示与 x_i 相同指标类型的所有情况的发生次数,亦等同于该类别所有事件的数量.

在实际的场景中,用户的行为特征可能会随着时间的推移而改变,这可能是由于自身行为习惯的改变,或者由外部因素(如开始负责不同的项目)所导致.这种现象称为概念漂移[26].因此,用户行为模式不能完全固定,需要随着用户习惯的变化而调整,以防止出现错误警报.为了减轻概念漂移的影响,我们采用指数权重衰减[27]方法,为不同时期的行为特征赋予不同的权重,逐渐忘记过时的习惯,而更加重视近期的习惯.权重的计算与当前时间到训练期的时间间隔成反比:

$$\omega_t=\mathrm{e}^{-\lambda\Delta t}, \quad (2)$$

其中,λ 是可以控制衰减速率的权重衰减因子,Δt 表示时间差.以 10 d 作为 1 个周期来执行权重衰减,Δt 等于时间间隔的周期数.则稀有度计算可改进为

$$r_{indicator_j}(x_i)=\begin{cases}0, \text{若 } x_i \text{ 未发生;}\\ \dfrac{\sum \omega_{t_a} N_{x_i}^{t_a}}{N_x}, \text{若 } x_i \text{ 发生,}\end{cases} \quad (3)$$

其中 a 为周期序号.

需要注意的是,训练期分值计算是将整个训练期间的行为习惯视为整体,不进行概念漂移的处理.

事件的稀有度得分由该事件所有维度指标的稀有度决定.为了平衡不同类别事件指标数量不同的区别,对分数进行了均值处理.定义某事件的事件稀有度为

$$s(e)=\frac{1}{m}\sum_{j=1}^{m} r_{indicator_j}, \quad (4)$$

其中 m 是该事件类的指标数量.

会话的异常性是由该会话中所有事件的稀有度得分决定的.为了减轻会话中由于事件类别不同和事件数量不同所造成的影响,取同一类别的所有事件得分的均值作为该事件类的得分.另外,存在一种可能的情况,在某个会话中某类事件没有发生,则根据计算流程,该类别分数计算结果为 0.对于用户不经常执行的某类事件,其在某次会话中不发生的可能性较高,而分数为 0 会严重影响后续的异常性判断.因此采用该类别事件不发生的频率作为其得分.为了进一步平衡某个事件类是否在会话中发生这 2 种情况的稀有度得分,将发生了此类事件的会话得分与事件类稀有度 r_{etype} 相乘.会话中某事件类 E_k 的稀有度分数为

$$s(E_k)=\begin{cases}1-r_{etype_k}, \text{若 } E_k \text{ 未发生;}\\ r_{etype_k}\times\dfrac{1}{n}\sum_{i=1}^{n} s(e_i), \text{若 } E_k \text{ 发生,}\end{cases}$$

其中,k 为事件类序号,n 为会话中该类事件的数量.

最后,依照式(5)计算 1 个会话的 7 个事件类分数.当会话结束,该会话的异常性分数可以立即计算得到,从而能够确保攻击发现的及时性.

2.4　检测算法

在计算出 1 个会话的分数后,利用检测算法来判断会话是否异常.对于实际场景,我们无法获得足够的带标签数据,因此采用有监督的分类算法是不合理的.本文采用异常值判断和多种无监督聚类算法进行异常会话的检测.算法的细节和各种算法的效果将分别在 3.2 节和 3.3 节中具体描述.对于检测结果为良性的会话,会话中的所有事件都将用于更新该用户的行为模式.对于检测结果为异常的会话,

则会触发警报.随后,我们对异常会话中的事件序列构建事件链,还原用户具体的操作并进一步精确判断窃密攻击.

应该注意的是,良性会话也可能被检测为异常,这是因为用户做了一些不常见但非恶意的操作.在这种情况下会进行及时反馈,并将此会话利用到用户行为模式更新中,以适应其行为习惯的变化.即使该事件仅仅是偶然事件,行为模式也不会被此类事件污染,因为该用户历史的行为特征依然在发挥作用,未来的行为也会不断更新到其个人行为模式中.

2.5 事件链构建

构建异常会话的事件链有 2 个好处:首先,可以得到用户具体的恶意操作,而不是仅仅知道他是恶意的,这也是目前相关研究工作的不足;其次,由检测算法确定的异常会话可能是良性的,因为其中存在一些罕见但良性的事件,从而导致误报,通过对事件链进行构建和分析,并与窃密攻击模式进行对比,可以做出更准确的判断.

对于执行窃密攻击的用户,有可能在 1 个会话中只进行恶意操作,也可能执行大量正常事件以隐藏恶意活动.因此,使用会话中的所有事件来构建事件链会造成时间和资源的浪费.我们利用之前计算得到的会话中每个事件的稀有度得分进行优先级排序,稀有度越低,则事件的优先级越高.由此,会话中的异常事件具有更高的优先级,从而先于正常事件用于事件链构建.在实时监控中,事件分数已经进行了均值处理,因此不会受到不同事件类别指标数量的影响,可以进行比较和排序.

根据事件间的时间关系和信息流关系构建事件链,时间关系即时间先后顺序,信息流关系即 2 个事件之间发生了明确的信息流动,例如,文件操作事件与带附件的电子邮件发送之间存在信息流动,则具有依赖关系.对于存在明确信息流依赖关系的事件,以信息流关系串联 2 个事件;对于不存在明确信息流关系的事件,则按照时间顺序建立连接.

事件链的具体构建步骤为:

1) 将异常会话中除登入、登出以外的事件按照稀有度分值从低到高排列;

2) 依次读取事件,利用事件属性(如时间、名称、操作等)建立节点;

3) 从第 2 个事件节点开始,与已有节点存在依赖关系的,根据信息流关系建立连接;

4) 当连续 n 个事件无法与已有节点建立连接时,停止事件读取,$n=\min(5,count_{rest\_events})$;

5) 已有节点未与其他节点关联的,按时间顺序建立连接;

6) 建立登入、登出事件节点,分别作为事件链的头和尾,按照时间顺序与其他节点串联.

接下来,将构建完成的事件链与窃密攻击模式进行对比,以精确发现窃密攻击的发生.本文利用专家知识建立窃密攻击模式.通常,窃密攻击的数据渗出过程主要有 3 种方式:通过可移动设备(如 U 盘)拷贝、通过电子邮件发送、以及上传到互联网.因此,我们关注从文件操作到可移动设备、文件操作到电子邮件发送、文件操作到网络上传这 3 种信息流模式,能够匹配特定模式的事件链被确定为窃密攻击.

3 实验与结果

3.1 数据集

使用卡内基梅隆大学的 CERT 内部威胁数据集 r6.2[28]进行实验.这是一个仿真数据集,模拟了完整的企业内网场景,记录了 4 000 名用户在 516 d 内的行为日志,包括登入登出、文件操作、电子邮件操作、可移动设备连接和网络活动.此外,数据集还提供了员工心理测量、部门隶属关系等信息.本文只利用用户事件日志,忽略该数据集提供的其他类型数据,以保证方法的通用性.该数据集中有 5 个攻击场景,其中的 2 个场景:用户 ACM2278、用户 CDE1846 涉及到窃密攻击的完整事件日志,因此,本文以这 2 个场景作为检测对象.为了防止另外 3 个攻击场景对算法检测结果的影响,本文将其排除在数据集之外.选取每个用户前 60 d 的行为日志作为训练期,进行用户行为习惯的学习和正常行为模式的建立,因此日志记录总天数不多于 60 d 的用户也被排除.本文以检测 60 d 后的日志中发生了窃密攻击的会话为目标.则筛选后的数据集详细情况如表 3 所示:

Table 3 Ground Truth of Dataset

表 3 数据集详细信息

类别	总体数量	恶意数量
用户	3 963	2
会话	1 372 639	11

对 2 个恶意用户进行简单的介绍:

1) 用户 ACM2278 以往从不使用可移动设备,也从不在下班后工作.他开始在下班后登录,使用可移动设备并将数据上传到维基解密网站,不久后离职.日志记录了他 2 次完整的窃密行为.

2) 用户 CDE1846 在 3 个月内越来越频繁地登入到另一个用户的计算机并访问感兴趣的文件,并将文件通过电子邮件发送到自己的邮箱.日志中记录了他 9 次完整的窃密行为.

3.2　实验设置

本文使用异常值判断和 4 种聚类算法来检测异常会话.

首先采用拉依达准则(pauta criterion)进行异常值判断,该方法能够以 99.7%的置信概率检测出异常值.在这里,将 1 个会话中的 7 个分数相加作为该会话得分.一般来说,用户行为习惯比较稳定,因此每个会话的得分也会趋于稳定,而异常会话会因事件的稀有性得分较低而呈现偏低的异常取值.利用用户前 60 d 的事件日志计算正常会话得分范围,并根据拉依达准则计算异常值阈值.由于分数衡量的是稀有性,分数越低越异常,因此只取区间下限作为异常值阈值 $\varepsilon=\mu-3\delta$,将得分低于该阈值的会话判定为异常会话.

为了使用聚类算法,将 1 个会话的 7 个分数转换成 7 维向量.由于正常用户有稳定的行为习惯,因此向量应彼此相似,则向量之间的距离应该接近;而恶意会话的向量应该与正常向量明显不同,在距离上应远离其他向量.聚类集群的数目是预先未知的,因此类似 K-Means 这类需要设置集群数量的算法无法解决这个问题.另外,每个用户的正常行为可能不止 1 个模式,因此一类支持向量机算法也不适用.本文采用均值漂移算法(Meanshift)、具有噪声的基于密度的聚类(density-based spatial clustering of applications with noise, DBSCAN)算法、孤立森林(Isolation Forest)算法、局部异常因子(local outlier factor, LOF)算法这 4 种聚类算法,它们都不需要提前设置集群的数量,且能够处理多集群的场景.

1) Meanshift 是一种非参数算法,通过移动质心来寻求密度函数趋于最大值,并根据质心和带宽确定聚类结果.在本文实验中,质心和带宽通过用户前 60 d 的会话日志训练获得.如果新会话的向量不能聚类到任何已有集群中,即其与所有质心之间的距离均大于带宽,则将被检测为异常会话.

2) DBSCAN 是一种基于密度空间的算法,能够把具有足够高密度的区域划分为 1 个集群,且可以生成任意形状的集群.异常会话与正常会话不相似,因此通常处于低密度区域,无法与数量占大多数的正常会话聚为 1 类,因此能够被该算法检测出来.

3) 孤立森林算法的核心思路是构建树型结构对数据集进行划分,在树的每一层随机选择特征,并根据特征的取值将不同的样本划分到不同的分支上,迭代划分过程,直到数据集不可再分或达到限定最大深度.树型结构建立完成后,计算每个样本点的深度以衡量其异常程度.由于正常样本点彼此相似,需要用更多的特征来划分,而异常样本点存在明显差异,很快就会被划分到不同的分支上.因此,样本点的深度越小,则异常程度越高.

4) LOF 算法通过比较每个样本点与其领域点密度来判断异常性,因此称为"局部"异常因子算法.具体来说,计算某样本点的邻域样本点所处位置的平均密度与该样本点所在位置的密度的比值,该比值越大,则说明该样本点所处位置的密度越低,则异常程度越大.

表 4 总结了 4 种聚类算法得到最优结果的具体参数设置:

Table 4　Parameter Settings of Clustering Algorithm

表 4　聚类算法参数设置

算法	参数
Meanshift	$quantile=1$
DBSCAN	$eps=0.85, min\_samples=35$
Isolation Forest	$n\_estimators=10, contamination=0.001$
LOF	$n\_neighbors=50, contamination=0.001$

3.3　实验结果

本节描述 5 种检测算法对于异常会话检测的效果.首先,利用准确率评估各算法判断正确的结果占总数的比例.另外,对于攻击发现,必须保证没有漏报,同时可以容忍一定程度的误报,因此,我们进一步利用真阳性率(true positive rate, TPR)和假阳性率(false positive rate, FPR)分别评估算法的漏报、误报情况.具体来说,TPR 衡量恶意会话被检测为恶意的比率,FPR 衡量良性会话被错误判断为恶意会话的比率.

表 5 汇总了 5 种检测算法各自的表现,包括准确率、TPR 和 FPR.可以看到,5 种算法的检测准确率普遍较高,均达到 98%以上的水平,其中 DBSCAN 和 LOF 算法达到了 99%以上的准确率.另外,5 种算法的 TPR 都达到 1,意味着所有的恶意会话都被成功检测,没有漏报发生.而 5 种算法的 FPR 水平都较低,说明误报率处于可以接受的范围内,但彼此之间有所差别.首先,利用拉依达准则进行异常值判断的 FPR 处于相对较高的水平,经过对实验结果中

各异常会话分值的分析,很多会话分值以微小的差距低于异常分数阈值而被判断为异常,因此误报率相对较高.其次,Meanshift 算法的误报率也较高,这出于其算法机制的影响.该算法根据质心和带宽确定聚类结果,那么每个集群的形状都是规则的圆形,使其不能很好得适应更复杂的集群形状与集群分布,导致了较高的误报率.孤立森林算法具有随机选择特征和计算节点深度的判断机制,对于异常会话,存在恶意行为的特征维度被选择的顺序将影响到正常、异常样本节点的深度差异,因而影响到检测的准确性和稳定性.最后,DBSCAN 算法和 LOF 算法都是基于密度的聚类算法,并且可以很好地处理形状不规则的集群,因此都取得了更低的 FPR 值.

Table 5 Results of Anomalous Session Detection

表 5 异常会话检测结果 %

算法	真阳性率	假阳性率	准确率
Pauta Criterion	100	1.24	98.76
DBSCAN	100	0.11	99.89
Isolation Forest	100	1.22	98.78
LOF	100	0.33	99.67
Meanshift	100	1.97	98.03

在检测结果为异常的会话中,存在着一些非恶意的会话,这些会话中的用户事件与该用户的历史行为习惯不相匹配,但并非出于恶意目的.一些可能的场景包括,某用户因为突发任务在非正常的时段登入了计算机、从不使用可移动设备的某用户由于项目交接从可移动设备中拷入了文件等.为了进一步提高恶意会话检测和窃密攻击发现的精确程度,降低误报并避免不必要的资源浪费,本文提出了事件链构建和窃密攻击模式匹配方法.

3.4 事件链构建

我们对检测算法判断为异常的会话进行事件链构建的处理,即将会话中的事件序列依据相互间的时间和信息流依赖关系建立关联,从而构成事件链,以还原用户所实施的具体操作.另外,由于本文的目标是进行窃密攻击的及时发现,我们还将事件链与窃密攻击模式进行了对比,将能够与窃密攻击模式匹配的事件链判定为窃密攻击,实现更低的误报率和更精确的攻击发现.

对 5 种算法检测得到的异常会话分别进行事件链构建的实验,并记录事件链构建前后被错误判断为恶意的良性会话数量,即假阳性样本量,以更直观地展示效果的提升.结果如图 3 所示,对于每种算法,假阳性样本量都有了明显的下降,这说明很多不包含恶意窃密操作的会话事件链被重新判定为正常.

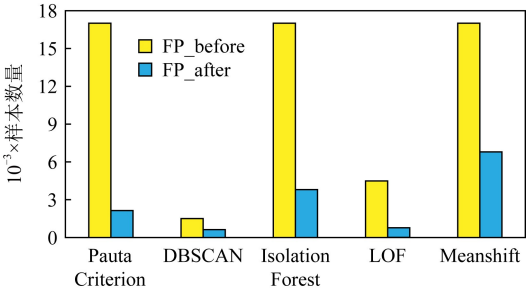

Fig. 3 Comparison of FP before and after event chain construction

图 3 事件链构建前后的假阳性样本数量对比

此外,我们再次统计检测结果的准确率、TPR 和 FPR 值,以观察事件链构建和窃密攻击模式匹配所带来的结果改善,统计结果如表 6 所示.准确率在之前的高标准基础上有了进一步提升;TPR 依然保持为 1,意味着所有存在窃密攻击行为的恶意会话都没有漏报;而 FPR 也都明显降低,均小于 0.005.

Table 6 Results of Anomalous Session Detection After Event Chain Construction

表 6 事件链构建后的异常会话检测结果 %

算法	真阳性率	假阳性率	准确率
Pauta Criterion	100	0.16	99.84
DBSCAN	100	0.05	99.95
Isolation Forest	100	0.28	99.72
LOF	100	0.06	99.94
Meanshift	100	0.49	99.80

由此可以说明,对会话事件序列进行事件链构建和窃密攻击模式匹配在提高准确率、降低误报率同时保证无漏报的方面发挥了积极的作用.

提高窃密攻击发现的精确程度是事件链构建的目的之一.更重要的是,可以自动将用户的行为链还原出来,直观清晰地掌握用户在此会话中进行了什么样的恶意操作,该操作发生的位置、时间等属性,以及各项操作之间的依赖关系,以协助安全管理人员进行攻击分析、溯源及取证.

从 2 个恶意用户 ACM2278 和 CDE1846 所实施了窃密攻击的会话中各取 1 个会话为例进行事件链的构建,并将构建的图形化结果进行展示和分析.

用户 ACM2278 的事件链如图 4 所示.在该会话中,用户在凌晨 1 点登入自己的计算机;然后连接

可移动设备到计算机上,从中拷贝了 3 个文件,将其逐一上传到维基解密网站以渗出窃取的文件;上传完毕之后,他断开可移动设备连接,登出了计算机.在该案例中,用户在非正常工作时间登入计算机并执行了一系列操作,他连接了很少使用的可移动设备,并访问了从未访问过的维基解密网站,这一系列稀有的事件使得各类检测算法都将其判断为异常会话.从事件间信息流依赖关系分析,该会话事件链能够与通过网络上传的窃密渗出模式匹配,从而被精确判定为窃密攻击.

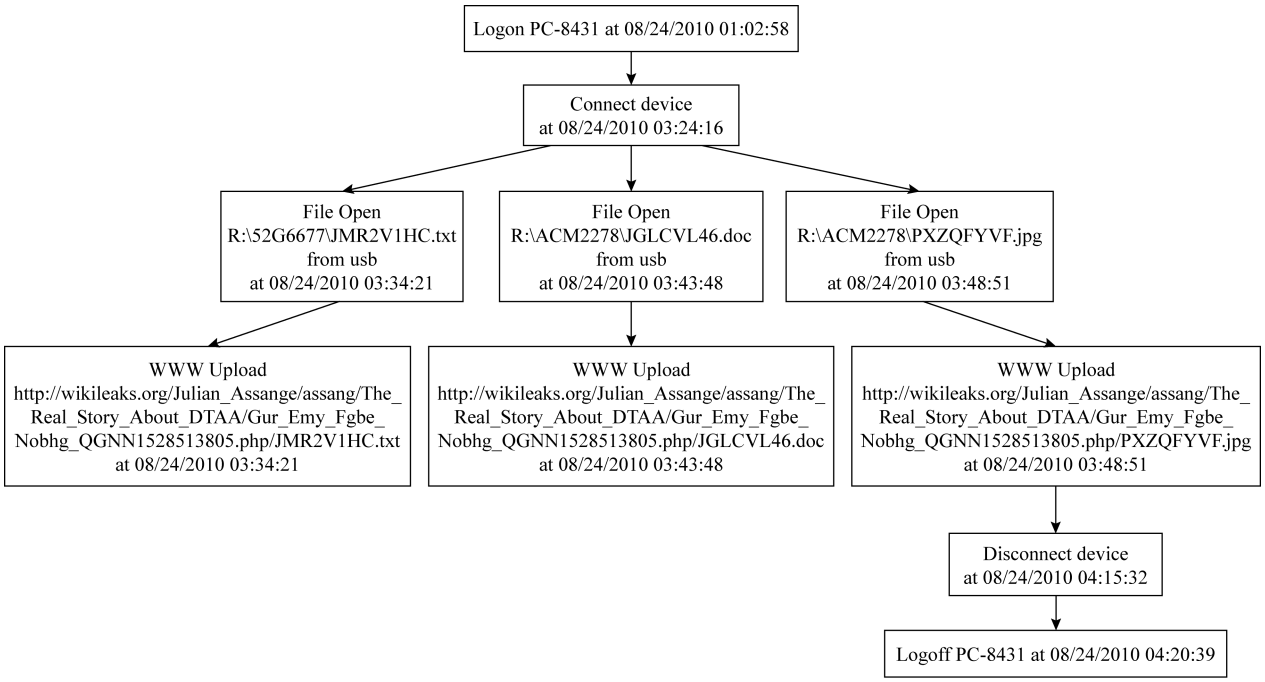

Fig. 4　Exfiltration attack event chain construction of user ACM2278

图 4　用户 ACM2278 的窃密攻击事件链构建

用户 CDE1846 的事件链如图 5 所示.在该会话中,用户在短短的 11 min 内实施了窃密操作.首先,他登入计算机,访问了几个文件,将这些文件通过电子邮件发送了出去,随即登出了计算机.在这一案例中,该用户登入了不属于自己的计算机,这在他历史的行为中非常罕见,然后他将几个文件通过带附件

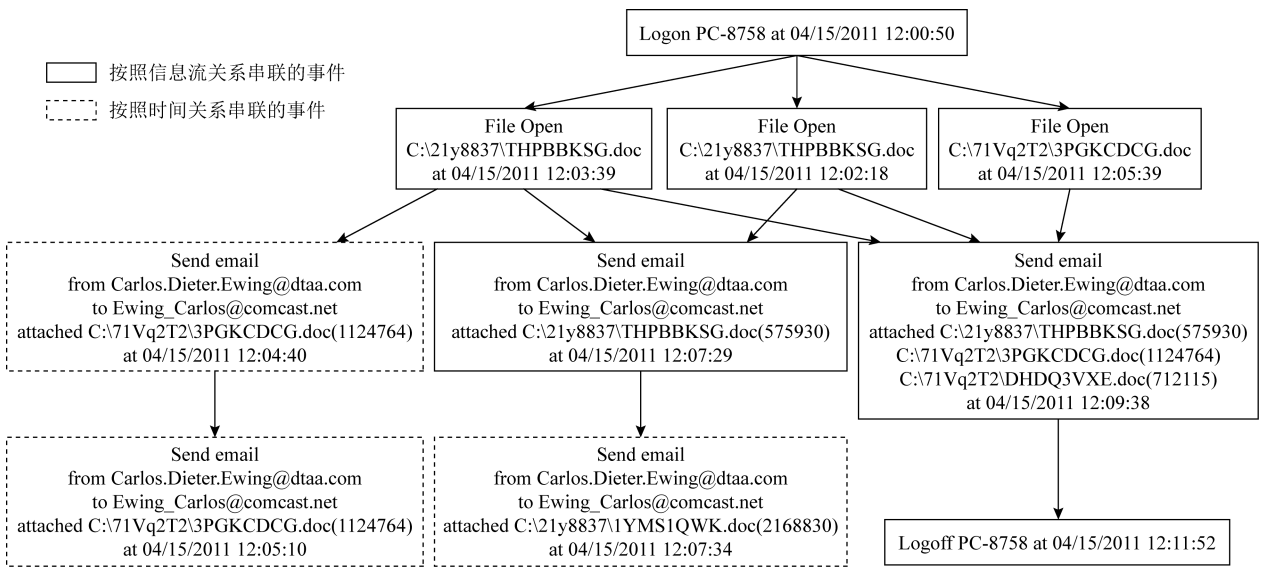

Fig. 5　Exfiltration attack event chain construction of user CDE1846

图 5　用户 CDE1846 的窃密攻击事件链构建

的邮件发出,这也是稀有程度较高的事件,因此被检测算法判定为恶意.该事件链符合通过邮件发送这一窃密渗出模式,因此成功被判定为窃密攻击.注意在左下角的 3 个虚线事件框,其稀有度较高,但无法与其他事件建立信息流依赖关系,因此按照时间顺序进行了关联,我们认为这可能是由于数据集日志记录的缺失导致.

在图 4、图 5 描述的 2 个案例会话中,都有与窃密攻击行为无关的网页浏览或文件操作事件,但因其稀有度分值较高、优先级更低且无法与其他事件建立信息流关联,因此没有进入事件链的构建.

4 总 结

本文提出了一种无监督的方法以实现窃密攻击的及时发现.利用用户事件,基于用户自身的正常行为习惯建立可更新的行为模式,利用无监督算法检测当前行为与正常行为模式的偏差,从而发现异常.以会话为单位的检测满足了攻击发现的及时性.另外,还提出构造异常事件序列的事件链,在更精确的判断窃密攻击的同时,还原用户具体的窃密操作.在卡内基梅隆大学 CERT 内部威胁数据集上的实验证明,本文提出的方法能够以无漏报、低误报的水平实现窃密攻击的及时发现和行为还原.未来研究计划应用更多的用户信息和属性来扩展用户行为模式,并尝试使用深度学习技术来探索更多的时间和空间特征,还计划使用概率模型来提升窃密攻击事件链的判断.

参 考 文 献

[1] Verizon. 2020 Databreach investigations report [EB/OL]. [2020-10-05]. https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf

[2] Ponemon. 2020 cost of insider threats: Global report [EB/OL]. [2020-10-05]. https://www.observeit.com/ponemon-report-2020-cost-of-ins ider-threats-global-cyberwire/

[3] Pohly D J, McLaughlin S, McDaniel P, et al. Hi-Fi: Collecting high-fidelity whole-system provenance [C] //Proc of the 28th Annual Computer Security Applications Conf. New York: ACM, 2012: 259-268

[4] Pasquier T, Han Xueyuan, Goldstein M, et al. Practical whole-system provenance capture [C] //Proc of the 2017 Symp on Cloud Computing. New York: ACM, 2017: 405-418

[5] Pasquier T, Han Xueyuan, Moyer T, et al. Runtime analysis of whole-system provenance [C] //Proc of the 2018 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 1601-1616

[6] Wang Fei, Kwon Y, Ma Shiqing, et al. Lprov: Practical library-aware provenance tracing [C] //Proc of the 34th Annual Computer Security Applications Conf. New York: ACM, 2018: 605-617

[7] Dong Boxiang, Chen Zhengzhang, Wang Hui, et al. GID: Graph-based intrusion detection on massive process traces for enterprise security systems [EB/OL]. 2016 [2020-01-09]. https://arxiv.org/pdf/1608.02639.pdf

[8] Liu Yushan, Zhang Mu, Li Ding, et al. Towards a timely causality analysis for enterprise security [C/OL] //Proc of the 25th Network and Distributed System Security Symp (NDSS). Rosten: The Internet Society, 2018 [2019-09-16]. http://www.princeton.edu/~pmittal/publications/priotracker-ndss 18.pdf

[9] Milajerdi S M, Gjomemo R, Eshete B, et al. HOLMES: Real-time APT detection through correlation of suspicious information flows [C] //Proc of 2019 IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2019: 1137-1152

[10] Han Xueyuan, Pasquier T, Bates A, et al. UNICORN: Runtime provenance-based detector for advanced persistent threats [C/OL] //Proc of the 27th Network and Distributed System Security Symp (NDSS). Rosten: The Internet Society, 2020 [2020-08-17]. https://www.ndss-symposium.org/wp-content/uploads/2020/02/24046-paper.pdf

[11] Le D C, Zincir-Heywood N, Heywood M I. Analyzing data granularity levels for insider threat detection using machine learning [J]. IEEE Transactions on Network and Service Management, 2020, 17(1): 30-44

[12] Ferreira P, Le D C, Zincir-Heywood N. Exploring feature normalization and temporal information for machine learning based insider threat detection [C] //Proc of the 15th Int Conf on Network and Service Management (CNSM). Piscataway, NJ: IEEE, 2019

[13] Aldairi M, Karimi L, Joshi J. A Trust aware unsupervised learning approach for insider threat detection [C] //Proc of the 20th IEEE Int Conf on Information Reuse and Integration for Data Science (IRI). Piscataway, NJ: IEEE, 2019: 89-98

[14] Jiang Jianguo, Chen Jiuming, Gu Tianbo, et al. Warder: Online insider threat detection system using multi-feature modeling and graph-based correlation [C] //Proc of 2019 IEEE Military Communications Conf (MILCOM). Piscataway, NJ: IEEE, 2019

[15] Ye Xiaoyun, Hong S S, Han M M. Feature engineering method using double-layer hidden Markov model for insider threat detection [J]. International Journal of Fuzzy Logic and Intelligent Systems, 2020, 20(1): 17-25

[16] Dahmane M, Foucher S. Combating insider threats by user profiling from activity logging data [C] //Proc of the 1st Int Conf on Data Intelligence and Security (ICDIS). Piscataway, NJ: IEEE, 2018: 194-199

[17] Le D C, Zincir-Heywood N, Heywood M I. Analyzing data granularity levels for insider threat detection using machine learning [J]. IEEE Transactions on Network and Service Management, 2020, 17(1): 30-44

[18] Jiang Jianguo, Chen Jiuming, Gu Tianbo, et al. Anomaly detection with graph convolutional networks for insider threat and fraud detection [C] //Proc of 2019 IEEE Military Communications Conf (MILCOM). Piscataway, NJ: IEEE, 2019: 109-114

[19] Gayathri R G, Sajjanhar A, Xiang Yong. Image-based feature representation for insider threat classification [J]. Applied Sciences, 2020, 10(14): 4945-4961

[20] Wei Renzheng, Cai Lijun, Yu Aimin, et al. AGE: Authentication graph embedding for detecting anomalous login activities [C] //Proc of 2019 Int Conf on Information and Communications Security. Berlin: Springer, 2019: 341-356

[21] Liu Fucheng, Wen Yu, Zhang Dongxue, et al. Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise [C] //Proc of the 2019 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2019: 1777-1794

[22] Zhang Shenglin, Li Dongwen, Sun Yongqian, et al. Unified anomaly detection for syntactically diverse logs in cloud datacenter [J]. Journal of Computer Research and Development, 2020, 57(4): 778-790 (in Chinese)
(张圣林, 李东闻, 孙永谦, 等. 面向云数据中心多语法日志通用异常检测机制[J]. 计算机研究与发展, 2020, 57(4): 778-790)

[23] Liu Liu, Chen Chao, Zhang Jun, et al. Unsupervised insider detection through neural feature learning and model optimisation [C] //Proc of 2019 Int Conf on Network and System Security. Berlin: Springer, 2019: 18-36

[24] Liu Liu, Chen Chao, Zhang Jun, et al. Insider threat identification using the simultaneous neural learning of multi-source logs [J]. IEEE Access, 2019, 7: 183162-183176

[25] Yuan Fangfang, Cao Yanan, Shang Yanmin, et al. Insider threat detection with deep neural network [C] //Proc of 2018 Int Conf on Computational Science. Berlin: Springer, 2018: 43-54

[26] Tsymbal A. The problem of concept drift: Definitions and related work [J]. Computer Science Department, Trinity College Dublin, 2004, 106(2): 58-64

[27] Klinkenberg R. Learning drifting concepts: Example selection vs example weighting [J]. Intelligent Data Analysis, 2004, 8(3): 281-300

[28] Glasser J, Lindauer B. Bridging the gap: A pragmatic approach to generating insider threat data [C] //Proc of 2013 IEEE Workshops on Security and Privacy. Piscataway, NJ: IEEE, 2013: 98-104

Feng Yun, born in 1993. PhD candidate. Her main research interests include cyber security, cyber-attacks discovery, attribution and forensic.
冯　云,1993 年生.博士研究生.主要研究方向为网络安全、网络攻击发现和溯源取证.

Liu Baoxu, born in 1972. PhD, professor, PhD supervisor. His main research interests include cyber security, threat intelligence, cyber-attacks attribution and forensic.
刘宝旭,1972 年生.博士,教授,博士生导师.主要研究方向为网络安全、威胁情报、网络攻击溯源取证.

Zhang Jinli, born in 1989. PhD candidate. Her main research interests include cyber security, cyber-attacks attribution.
张金莉,1989 年生.博士研究生.主要研究方向为网络安全、网络攻击溯源取证.

Wang Xutong, born in 1997. Master candidate. His main research interests include insider threat detection and machine learning.
汪旭童,1997 年生.硕士研究生.主要研究方向为内部威胁检测和机器学习.

Liu Chaoge, born in 1986. PhD, associate professor. His main research interests include malware, cyber-attacks attribution and Web security.
刘潮歌,1986 年生.博士,副研究员.主要研究方向为恶意代码、网络攻击追踪溯源和 Web 安全.

Shen Mingzhe, born in 1996. Master candidate. His main research interests include Web security, intrusion detection, forensic system.
申明喆,1996 年生.硕士研究生.主要研究方向为 Web 安全、入侵检测和取证系统.

Liu Qixu, born in 1984. PhD, professor, PhD supervisor. His main research interests include Web security and malware analysis.
刘奇旭,1984 年生.博士,教授,博士生导师.主要研究方向为 Web 安全和恶意代码分析.