ISSN 1000-0054 清华大学学报 (自然科学版) 2011 年 第 51 卷 第 10 期 14/25
CN 11-2223/N J Tsinghua Univ (Sci & Tech), 2011, Vol. 51, No. 10 1306-1311

DHT 网络 eclipse 攻击

邹 维[1,2], 张 缘[1,2], 张建宇[1,2], 周 模[1,2], 刘丙双[1,2]

(1. 北京大学 计算机科学技术研究所, 北京 100871; 2. 北京大学 互联网安全技术北京市重点实验室, 北京 100871)

摘 要: 分布式 Hash 表(distributed Hash table, DHT)是结构化对等网络的核心技术。实际 P2P 应用中, DHT 网络规模已经达到上千万节点,但是其安全问题仍然很多。eclipse 攻击是 DHT 网络中典型的安全威胁之一。本文介绍了 DHT 网络中 eclipse 攻击常见的攻击方法,总结归纳了近年来攻击检测和防御技术的研究工作进展,从适用场景、依赖条件和性能等方面对这些工作进行了对比分析,最后对未来的研究工作进行了展望。

关键词: 对等网络; 分布式 Hash 表(distributed Hash table, DHT); eclipse 攻击

中图分类号: TP 393 **文献标志码**: A

文章编号: 1000-0054(2011)10-1306-06

Survey of eclipse attacks on DHT networks

ZOU Wei, ZHANG Yuan, ZHANG Jianyu, ZHOU Mo, LIU Bingshuang

(1. Institute of Computer Science & Technology, Peking University, Beijing 100871, China;
2. Beijing Key Laboratory of Internet Security Technology, Peking University, Beijing 100871, China)

Abstract: The Distributed Hash table (DHT) is the core of structural peer-to-peer networks. There are millions of nodes in DHT networks in practical P2P applications. However, DHT networks still have many security problems which have not been resolved. The eclipse attack is a typical DHT network security threat. This paper describes the common eclipse attack methods and summarizes recent research on detection methods and eclipse attack defenses. The methods are compared in terms of applicable conductions, dependencies and performance. New issues and future directions for eclipse attacks are also given.

Key words: peer-to-peer (P2P); Distributed Hash Table (DHT); eclipse attack

分布式 Hash 表(distributed Hash table, DHT)是结构化对等网络的核心技术, DHT 网络以其分布式、自组织和可扩展等特性在 BitTorrent、eMule 等 P2P 文件共享应用中得到了广泛的使用。在 DHT 网络中,信息以〈key, value〉的形式映射到网络中的逻辑地址空间上,分散存储在键值 key 所对应的节点之中,所有节点存储的信息条目构成一个全网的分布式 Hash 索引表。每个 DHT 节点维护了一个在线邻居节点的集合,通过节点间的路由过程来定位信息搜索和发布的目标节点。DHT 网络的正常运转依赖于节点之间开放、自觉、有序的协作,但是这种节点之间的开放和信任很容易遭受破坏或被恶意利用。因此 DHT 网络中一直以来存在着比较多的安全问题[1]。

节点对网络的了解有限使得在这样一个开放的 DHT 网络中很难排除恶意节点的存在,因此恶意节点可以自由地加入到这个网络中,并且利用这种节点间有限的了解来发动攻击或者破坏网络的完整性,例如 sybil 攻击[2-4]、eclipse 攻击[5-9]、churn 攻击[10,11]等。

eclipse 攻击是 DHT 网络中一个典型的安全问题。本文总结归纳 DHT 网络中 eclipse 攻击常见的攻击方法,介绍近年来攻击检测防御技术的研究进展,并进行对比分析,对未来的研究方向进行展望。

1 eclipse 攻击

eclipse 攻击是指攻击者通过侵占节点的路由表,将足够多的恶意节点添加到其邻居节点集合中,从而将这个节点"隔离"于正常网络之外。这种攻击也被称为路由表毒化。当节点受到 eclipse 攻击时,节点的大部分对外联系都会被恶意节点所控制,由此恶意节点得以进一步实施路由欺骗、存储污染、拒绝服务以及 ID 劫持等攻击行为。因此, eclipse 攻击对 DHT 网络的威胁非常严重。

收稿日期: 2011-08-15

作者简介: 邹维(1964—),男(汉),重庆,研究员。
E-mail: zou_wei@pku.edu.cn

eclipse攻击和sybil攻击联系紧密。sybil攻击是指单个物理实体在覆盖网上产生大量不同的身份。显而易见,成功的sybil攻击可以使发动eclipse攻击变得更为容易。但是,即使防守方采取了有效的sybil防御措施,eclipse攻击依然可以被实施。在分布式的覆盖网中,节点定期地通过获得现有邻居节点的邻居表来发现新邻居。恶意节点可以利用这一点,将由恶意节点组成的邻居集合发送给正常节点,使得更多的恶意节点被加入到正常节点的路由表中。通过这种方法,小规模的拥有合法身份的恶意节点依然可以成功实施eclipse攻击。

2 检测恶意节点

Singh等[9]认为当拥有有限资源的敌手发动大范围的eclipse攻击时,恶意节点的入度一般要高于网络中节点的平均入度。基于这一事实,可以要求正常节点仅仅选择那些入度低于某一阈值的节点作为邻居节点。然而恶意节点仍然能够通过指向尽量多的正常节点来消耗正常节点的入度,因此限制每个节点的出度也是有必要的。

上述出入度限制可以通过一种分布式审计的方法来实现。每个节点 P 维护了一个backpointer集合,其中保存了将节点 P 作为邻居的所有节点的信息。节点 P 只会回应或转发来自它的backpointer集合中的节点的消息。另一方面,节点 P 还会周期性地请求它的邻居节点的backpointer集合。如果某个邻居节点返回的集合中节点的数量大于入度阈值,或者节点 P 不存在于返回的集合中,那么节点 P 会将这个邻居节点从邻居表中去除掉。类似的过程也要应用到节点 P 的backpointer集合的每个成员上,以保证它们的出度也要低于某个阈值。

这种方法能够发挥作用的前提条件是被审计的节点不能知道审计者的身份。为了实现这种匿名化,节点通过一些中间的节点来传递他们的审计消息,这些中间节点被称为anonymizer。审计时间的随机化可以使得对审计者身份的探测变得困难。

当节点 P 希望审计节点 Q 时,节点 P 首先从距离节点 Q 最近的节点中随机选择一些作为anonymizer节点。由于在正常情况下,网络中节点的ID是随机生成的,这些anonymizer节点中的恶意节点的比例的期望值应该与整个覆盖网中恶意节点的比例相等。

由于anonymizer节点可能是恶意的,因此需要对一个节点进行 n 次挑战,那些应答正确次数少于 k 的节点就认为是恶意的。一个正常的节点被误认为是恶意的概率(误报率)为

$$\sum_{i=0}^{k-1} \binom{n}{i} f^{n-i}(1-f)^{i}.$$

同时,一个恶意节点成功通过了审计检查的概率(漏报率)为

$$\sum_{i=0}^{k-1} \binom{n}{i} f^{i} + \binom{(1-f)c}{r} [(1-f)(1-c)]^{n-i}.$$

其中:c 是恶意节点回答挑战的概率,r 是恶意节点的backpointer集合大小与出入度阈值之比。当 $n=24$、$k=12$、$r \geq 1.2$ 并且假设 $f \leq 0.25$ 时,该方法的误报率约等于0.2%,恶意节点被检测出来的概率至少是95.9%。

当恶意节点的比例被设置为20%时,模拟实验结果如表1所示。表1表明入度限制策略对于维持较低比例的恶意路由表项是有效的。但是这种有效性会随着网络规模的增大或者入度阈值的放松而降低。

表1 在不同的入度阈值下正常节点路由表中恶意节点的比例

阈值	结点数量/%			
	1 000	5 000	10 000	20 000
16	24	24	24	24
32	24	29	31	37
48	24	31	35	45
64	24	33	38	48
无限制	24	35	42	50

对上述审计方法从入度分布、恶意节点探测、通信开销和误报率等方面进行了实验评估。当节点数量为2 000且审计参数 $n=24$、$k=12$ 时,实验表明这种审计方法能够成功地捕获所有入度过大的恶意节点;节点加入退出频率与审计频率之间呈正比关系;当审计周期为2 min时,审计开销(2 message/(node·s))、安全路由开销(0.2 message/(node·s))和维护开销(4.2 message/(node·s))都较低;经过10 h的持续模拟所产生的误报率仅为 10^{-3}。

这种防御方法的优点是它并不需要任何难于实现的中心化服务的辅助,并且不需要密码技术的支持。另外,它所应用的网络距离优化方法一定程度上遏制了eclipse攻击。但是这种方法也存在一些缺点。首先,只有当入度阈值设置较小时,这种方法才是效果显著的,而且在不存在攻击的情况下这种方法会带来搜索时间的延长。其次,这种审计的过程对于每个节点都是独立的,即每个节点都需要独

自发现所有的恶意节点,不能通过某种机制来交换恶意节点的信息。另外,对于针对某一小部分节点的局部 eclipse 攻击,这种限制入度的思想并不能识别出恶意节点,从而不能起到防御作用。

3 攻击容忍策略

3.1 路由表项冗余

Hildrum 和 Kubiatowicz[7]提出了一种通过在 Pastry[12]和 Tapestry[13]路由表中添加冗余表项的方法来减缓 eclipse 攻击的危害。这种方法的前提假设是网络距离(时延)的测量必须是可信的。对于路由表的第 X 行、第 Y 列,填入 r 个前 X 位相同、第 $X+1$ 位为 Y 且网络距离最近的邻居节点。这种方法能够抵抗 eclipse 攻击的原因是当恶意节点的比例足够小时,很难在大量的正常节点中占据网络距离最近的那些位置。路由表中表项的冗余增加了恶意节点占据路由表的难度。

在 50 000 的节点规模且恶意节点比例为 50 % 时,实验结果显示路由表毒化比例会随着 r 的增加而减少。这种方法的主要优势是简单,但是它依赖于可信且稳定的网络距离测量机制,而且并没有提到如何在现实中实现这种机制。另外,实验中所用的底层网格拓扑结构的距离测量方法和真实环境中是不同的。根据 Singh 等[9]的观点,这种防御方法仅适用于小规模且节点充分分散的覆盖网络。

3.2 路由冗余

为了抵抗在 Pastry 和 Tapestry 等采用了网络距离优化的 DHT 协议中的 eclipse 攻击, Castro 等[2]提出了一种使用 2 个路由表进行路由的方法。一个路由表(优化路由表)基于网络延迟信息,由与节点 X 的 ID 前缀相同且与节点 X 的网络距离最短的节点组成;另一个路由表(验证路由表)由能够被验证并且不考虑网络距离的节点组成,表项 (l, d) 对应第 $l+1$ 位为 d、而其余位与 X 相同的 ID, 表项中包含的是距离这个 ID 最近的节点。在通常情况下,节点优先使用路由效率相对较高的优化路由表进行路由,在路由检测失败的时候则依靠验证路由表通过安全冗余路由来重新进行路由。

提出了一种路由失败检测方法来对基于优化路由表的路由结果进行检测。定义路由请求者与其邻居节点间的平均距离为 d_p, 路由目标 ID 与路由结果中所有节点之间的平均距离为 d_r, 当 $d_r < d_p \cdot \gamma$ 时,则判定路由失败,其中 γ 为调节误报率和漏报率的参数。这种检测方法主要针对的是恶意节点集中于路由请求者或副本节点附近的攻击类型,当足够多的恶意节点有针对性地靠近路由请求者或副本节点时,上述检测到的节点间平均距离就会被改变,从而导致误报率或漏报率的增加。

在路由检测失败的时候,节点使用一种冗余路由的方法来重新进行路由,即从多条路径向目标进行路由搜索。如果副本节点充分分散在整个 ID 空间中,那么这种冗余路由的效果是显著的。但是如果副本节点是分布在临近目标 ID 的一些位置上,那么这种冗余路由依然不能提升路由效果。

由于消息通过一条安全的路径到达目标的概率是 $(1-f)^{\log_{2^b} N}$。其中: 2^b 是路由表的列数, N 是整个网络的节点个数, f 是网络中恶意节点的比例。因此通过上述步骤路由查询请求到达所有正常副本节点的概率大致也等于这一概率。这里并没有具体说明如何来验证通过冗余路由获得副本节点集合是否完整,但是这部分的验证可以通过使用大多数投票的方法来完成。需要注意到的是冗余路径的数量可能会很大,因为单一的恶意节点就可以使经过它的所有路径完全失效。Hildum 和 Kubiatowicz[7]认为如果要使得成功路由的概率保持在一个常数水平,那么相互独立的路由路径的数量就需要以节点规模的多项式级增长。另外,这种冗余路由的效率非常依赖于优化路由表的使用频率。路由失败检测本身依赖于返回节点的平均距离的真实性,而这一计算距离结果很可能会被恶意节点控制,从而导致使用冗余路由的频率急剧增加。

由于 eclipse 攻击会严重毒化优化路由表,在一段时间后大部分的路由过程将不得不通过验证路由表来完成。这一过程的开销包含了使用已经被毒化的优化路由表进行路由以及路由失败检测的开销。在这种情况下,单独使用验证路由表进行冗余路由并且摈弃路由失败检测会是一种更好的方法。另外这种方法需要依赖一个集中式的分发节点 ID 的权威服务,这在分布式的覆盖网络中也是一个不提倡的做法,因为集中式的服务存在较高的脆弱性和安全性问题。但没有进行实验以从整体上评价这种方法的效果。

3.3 路由表重置

Condie 等[6]提出了一种通过诱发网络抖动(churn)来防御 eclipse 攻击的方法。使用双路由表的方法由于优化路由表的毒化程度会不断加剧而具有一定的脆弱性。解决上述脆弱性的办法是周

期性地将优化路由表重置为验证路由表并且在大部分情况下使用优化路由表进行路由操作。当优化路由表毒化程度缓慢增长时,这种方法是有效的。因此为了抑制毒化速度就必须限制路由表的更新频率,因为路由表更新操作是毒化加剧的主要源头。

为了避免攻击者发现路由表更新的规律,这里必须引入一种不可预测的 ID 分配方法。每次路由表重置时,节点获得一个新的随机 ID,将自己转移到不同的 ID 子空间中。如果正常节点持续地移动位置,恶意节点将很难在每次路由表重置后对其进行攻击。所有的节点不能同时进行路由表更新操作,因为这样会导致系统负载变得极其不稳定,而且恶意节点可以摸索路由表更新的规律,在恰当的时刻发起攻击来使得路由表毒化程度迅速恶化。因此,将所有节点按照节点 IP 前缀划分成多个组,通过交错地进行路由表更新可以避免上述攻击行为。

模拟实验结果表明:这种方法显著地减少了路由表毒化的比例,增加了路由成功的概率。当恶意节点比例超过 5 %时,与单独使用基于验证路由表的冗余路由效果相比,单独使用路由表重置方法具有更好的效果。在恶意节点比例是 15 %时,这 2 种方法的结合可以使路由成功的概率接近于 1,而单独使用路由表重置的结果近似是 0.35,单独使用冗余路由的结果近似是 0.15。在恶意节点比例是 25 %时,这种结合的方法使路由成功概率达到接近 0.8,而单独使用路由表重置和冗余路由的结果分别是 0.3 和 0.1。

这种方法在一定程度上能够有效防御 eclipse 攻击,但是引入节点的主动退出和加入机制会造成存储在网络上的数据不稳定。在每次路由表更新时,每个数据项至少会移动一次,至少增加了负载 $\Omega(k)$,k 是存储的键值数量。作者研究了系统的负载情况,但是仅仅考虑了更新、维护路由表以及探测节点失效的开销。对于很多应用,特别是在具备冗余存储的系统中,移动数据的开销占据带宽消耗的主要部分。此外,该方法还需要考虑支持在线可信 ID 生成服务的管理开销,这种服务的中心化特点可能不适用于分布式的 P2P 系统。

4 攻击限制策略

4.1 节点 ID 生成策略

4.1.1 基于验证服务的策略

Kademlia[14] 是在 P2P 网络中广泛使用的一种 DHT 协议,它已经被应用到多个商业文件共享系统中。Maccari 等[8] 提出一种通过外部验证服务来解决 Kademlia 网络中 eclipse 攻击问题的方法。通过模拟 1 000 000 节点规模的网络中 eclipse 攻击的情况,发现如果攻击者任意选择节点 ID,并且在目标 ID 发布之前就部署了邻近的恶意节点,那么攻击成功的概率几乎是 100 %。攻击者仅仅需要控制 8 个节点 ID 就可以达到这种效果。因此,一些学者提出基于用户的 IP 地址和端口号进行 Hash 来生成节点 ID,但是模拟实验表明如果攻击者拥有足够多的 IP 资源,eclipse 攻击依然能够在很大的概率下获得成功。上述实验表明:Kad 网络对多种攻击的脆弱性由以下 2 个因素造成,一是攻击者可以自由选择节点 ID;二是攻击者可以获得多个节点 ID。

通过使用验证服务对节点 ID 进行签名可以防止攻击者随意选择自己的节点 ID 并且生成大量节点。在一个新加入的节点初始化前,它需要向验证服务器发送一个 NodeIdRequest 消息来获得自己的节点身份。验证服务器完成用户身份的验证过程,然后将用户身份、用户的公钥以及节点 ID 进行绑定,生成标识 AuthId = Sign(NodeId ‖ UserId ‖ K^+ ‖exp, K^-),将其返回给请求节点。其中 NodeId 是验证服务器随机选取的,exp 是 NodeId 过期的时间戳,这个标识用验证服务器的私钥进行签名。验证服务器维护了 UserId 与 AuthId 之间的映射关系,因此所有来自于同一个 UserId 的后续的 NodeIdRequest 消息都将会返回之前生成的同一个 AuthId 信息,除非前一个生成的 AuthId 已经过期。这样可以防止一个攻击者生成多个节点。

定义了一种基于交换签名标志的节点交互协议来验证节点的身份。节点 A 与节点 B 的一次正常通信过程为:

I. $A \rightarrow B$: $NodeId_A$, N_1
II. $B \rightarrow A$: $NodeId_B$, N_2
III. $A \rightarrow B$: $AuthId_A$, $Auth_{AB}$, 请求消息
IV. $B \rightarrow A$: $AuthId_B$, $Auth_{BA}$, 应答消息

其中:N_1 和 N_2 是 2 个随机生成的 nonce,用于防御中间人攻击。签名标识 $Auth_{xy}$ 由节点 x 的 ID、消息的 Hash 值以及前面从节点 y 获得的 nonce 组成。

在 Kademlia 网络中,如果节点的路由表中存在空位,它就会将接收到的消息的发送者信息添加到路由表中。AuthId 的申请与交互协议的结合使用使得 2 个节点之间的通信必须经过验证,由于节点 ID 是

验证服务器随机选取的,攻击者无法控制自己的节点在 ID 空间中的位置,亦即无法侵占目标节点路由表的特定位置,也就无法成功实施 eclipse 攻击。

这种方法存在一定的局限性。首先,该方法依赖于集中式的身份验证服务,而集中式的服务存在可扩展性差、脆弱性以及安全性上的不足,不适用于分布式覆盖网络。其次,如果攻击者拥有足够多的资源,能够产生大量不同的用户身份,那么这种方法依然不能防止攻击者生成大量的恶意节点,而且攻击者可以通过使用不同的用户身份多次尝试获取节点 ID,从中选择合适的离目标较近的 ID 作为恶意节点的 ID 来实施 eclipse 攻击。

4.1.2 Cukoo 策略

Awerbuch 和 Scheideler[5] 引入了在节点空间[0,1)范围中区域(region)的概念。节点的 k-region 定义为地址空间中大小为 k/n 并且覆盖节点 ID 的区间,n 是网络中正常节点的数量。当一个节点加入网络时,使用可验证的秘密共享方法来保证节点 ID 的随机性。使用这种秘密共享方法生成随机 ID 的前提假设是参与 ID 生成的节点中,所有正常节点必须互相了解,并且其中恶意节点的比例不能超过一定的阈值。然而这种前提假设在实际的网络中可能很难保证,从而使得上述方法无法保证其实用性。一个恶意节点可以连续尝试加入和离开系统,直到它收到一个合适的节点 ID 为止。这种行为可以使得攻击者在多个区间里注入恶意节点,从而污染正常节点的路由表。提出一种被称为 cuckoo 策略的协议来进行防御。协议规定当一个新节点 X 加入覆盖网时,所有存在于节点 X 的 k-region 中的节点必须离开系统,并且使用新的随机节点 ID 重新加入网络。证明了该协议能够保证所有区间中的节点数量保持均衡,并且只要 $\varepsilon<1-1/k$,在每个区间中正常节点的数量将会占据绝大多数,ε 代表恶意节点与正常节点的比例。

使用 d 维的 de Bruijn 图来对覆盖网进行建模,使用向量$\{1,0\}^d$表示节点,当 d 趋近于无穷时节点。节点 $X=\sum_{i=1}^{d}x_i/2i\in[0,1]$。节点 X 的 quorum 区域 R_x 被定义为大小近似于 $(Y\log n)/n$ 并且包含 X 的区域,其中 Y 是一个大于 1 的常数。对于任意一个节点集合 V,每个属于 V 的节点 v 维护了一个节点信息列表,列表中的所有节点的 quorum 区域至少包含节点集合$\{v, v/2, (1+v)/2, 2v \bmod 1, (2v-1) \bmod 1\}$之中的一个节点。当一个节点 X 加入或离开时,只有那些与集合$\{x, x/2, (1+x)/2, 2x \bmod 1, (2x-1) \bmod 1\}$相交的 quorum 区域才会受到影响,这些受影响的区域数量为 $O(\log n)$。

4.2 基于区间的路由协议

Awerbuch 和 Scheideler[5] 的方法除了对节点加入作出限制,还对路由协议作出了改进。为了从节点 X 发送一个消息到节点 Y,X 用二进制表示为 $(x_1\ x_2\ \cdots x_{\log n})$(这里只关注节点用二进制表示的前 $\log n$ b),Y 用二进制表示为 $(y_1\ y_2\ \cdots y_{\log n})$,消息必须沿着包含 $(x_2\ x_3\ \cdots x_{\log n} y_1)$,$(x_3\ x_4\ \cdots x_{\log n} y_1\ y_2)$ 等的 quorum 区域进行转发,直到到达包含 $(y_1\ y_2\ \cdots y_{\log n})$ 的 quorum 区域为止。通过引入 shuffle graph 避免可能由恶意节点产生的假消息所造成的通信阻塞。搜索操作允许对每个搜索请求尝试多次,并且使用简单的阈值来控制每次尝试的拥塞。

通过数学推导证明了上述防御方法的有效性,但是没有进行模拟实验加以验证,所以实际的可操作性以及实用性并没有得到明确的说明。另外,没有说明在这种方法中节点的路由表应该如何进行维护,也没有对方法的系统负载和开销进行分析。

5 结 论

防御 eclipse 攻击的方法需要在性能和复杂度之间进行权衡。成功的防御方法通常只能保证路由表中恶意表项的比例等于覆盖网络中恶意节点的比例。这就意味着上述介绍的防御技术并不足以保证正常的 DHT 网络操作,通常还需要结合其他技术来提升 DHT 网络操作的正确性,例如冗余路由等。

上述的各种防御方法的前提假设、适用场景等都不尽相同,如表 2 所示。Singh 等提出的方法对于全网范围的攻击能够起到一定效果,但是对于局部特定

表 2 各种 Eclipse 攻击防御方法对比

方法	依赖条件	适用场景
检测恶意节点	无	Pastry 等全网范围的攻击
路由表项冗余	可信的网络距离测量	Pastry 和 Tepastry 小规模分散网络
路由冗余	中心化 ID 验证服务	Pastry、Tepastry、Chord[15]、CAN[16]
基于验证的节点 ID 生成策略	中心化随机数服务	Kademlia
路由表重置	网络存储数据规模较小	Pastry 等轻数据负载
基于区间的路由协议	无	提出新的路由协议

位置的 eclipse 攻击无能为力。Hildrum 等的方法适用于 Pastry 和 Tepastry 网络结构,其前提是网络距离的测量必须是可信的,最终它只能适用于小规模并且节点充分分散的覆盖网络。Castro 等和 Maccari 等的方法都需要中心化服务的辅助才能实现节点 ID 生成的随机性,而中心化服务存在一定的脆弱性和安全性问题,尤其在分布式系统中更为突出。Condie 等提出的方法虽然与 Castro 等的方法相近,但是由于节点会频繁地退出和重新加入,因此这种方法并不适用于网络上存储大量数据的应用。

针对 eclipse 攻击的研究工作可以从攻击的检测、容忍和抵御着 3 个角度来进行。上述介绍的各种方法都集中在了攻击的检测和容忍这 2 个方面,而目前还没有能够真正有效抵御 eclipse 攻击的方法。

eclipse 攻击可以分为大范围攻击和局部攻击,Singh 等[9]发现大范围攻击的一个本质特点就是恶意节点的入度超过了节点的平均入度,因此利用这一特点就可以检测出恶意节点。与大范围攻击的特点不同,局部攻击需要恶意节点必须具备足够的能力来获得特定 ID,从而能够"包围"特定区域,然而在众多的节点 ID 分配方法中没有一种能够完全防止敌手生成特定 ID,只要敌手拥有足够多的资源,就可以获得想要的 ID。

eclipse 攻击能够成功的原因之一是节点不能区分其路由表中的节点是否可信,引入社会网络中的信任机制可能是一个不错的选择,节点可以通过其信任的朋友来获得正确的路由信息、区分恶意节点。另外通过对节点历史信息的统计也可以在一定程度上判断出节点是否可信。

参考文献 (References)

[1] Urdaneta G, Pierre G, Steen M. A survey of DHT security techniques [J]. *ACM Computing Surveys*, 2009, **43**(2): 8.

[2] Castro M, Druschel P, Ganesh A, et al. Secure routing for structured Peer-to-Peer overlay networks [C]// Proceedings of the 5th Symposium on Operating Systems Design and Implementation. New York: Association for Computing Machinery, 2002: 299 - 314.

[3] Douceur J. The Sybil attack [C]// Proceedings of the 1st International Workshop on Peer-to-Peer Systems. London, UK: Springer-Verlag, 2002: 251 - 260.

[4] Dinger J, Hartenstein H. Defending the Sybil attack in P2P networks: Taxonomy, challenges, and a proposal for self-registration [C]// Proceedings of the 1st International Conference on Availability, Reliability and Security. Washington DC, USA: IEEE Computer Society Press, 2006: 756 - 763.

[5] Awerbuch B, Scheideler C. Towards a scalable and robust DHT [C]// Proceedings of the 18th Annual ACM Symposium on Parallelism in Algorithms and Architecture. New York, USA: Association for Computing Machinery, 2006: 318 - 327.

[6] Condie T, Kacholia V, Sankararaman S, et al. Induced churn as shelter from routing table poisoning [C]// Proceedings of the 13th Annual Network and Distributed System Security Symposium. San Diego, USA: The Internet Society, 2006.

[7] Hildrum K, Kubiatowicz J. Asymptotically efficient approaches to fault-tolerance in Peer-to-Peer networks [C]// Proceedings of the 17th International Symposium on Distributed Computing Lecture. Berlin, Germany: Springer-Verlag, 2003: 321 - 336.

[8] Maccari L, Rosi M, Fantacci R, et al. Avoiding eclipse attacks on Kad/ Kademlia: An identity based approach [C]// Proceedings of the IEEE International Conference on Communications. Piscataway, USA: IEEE Press, 2009: 1 - 5.

[9] Singh A, Ngan T, Drushel P, et al. Eclipse attacks on overlay networks: Threats and defenses [C]// Proceedings of the 25th International Conference on Computer Communications. Barcelona, Spain: IEEE Press, 2006: 1 - 12.

[10] Rhea S, Geels D, Roscoe T, et al. Handling churn in a DHT [C]// Proceedings of the Annual Conference on USENIX Annual Technical Conference. Berkeley, USA: USENIX Association, 2004: 127 - 140.

[11] Godfrey, P, Shenker S, Stoica I. Minimizing churn in distributed systems [C]// Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York, USA: Association for Computing Machinery, 2006: 147 - 158.

[12] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems [C]// Rachid Guerraoui, ed. Proceedings of the IFIP/ ACM International Conference on Distributed Systems Platforms Heidelberg. London, UK: Springer-Verlag, 2001: 329 - 350.

[13] Zhao B, Huang L, Stribling J, et al. Tapestry: Are silient global-scale overlay for service deployment [J]. *IEEE Journal on Selected Areas in Communication*, 2004, **22**(1): 41 - 53.

[14] Maymounkov P, Mazieres D. Kademlia: A peer-to-peer information system based on the XOR metric [C]// Proceedings of the 1st International Workshop on Peer-to-Peer Systems. London, UK: Springer-Verlag, 2002: 53 - 65

[15] Stoica I, Morris R, Liben-Nowell D, et al. Chord: A scalable peer-to-peer lookup protocol for Internet applications [J]. *IEEE/ ACM Transactions on Networking*, 2003, **11**(1): 17 - 32.

[16] Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network [C]// Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York, USA: Association for Computing Machinery, 2001: 161 - 172.