集群软件心跳机制的安全性设计

谭毓安（北京理工大学计算机科学与工程系 100081）

摘要：在集群系统中，心跳是结点之间的通信手段。本文从心跳数据的鉴别和抵御重发攻击两个方面讨论了心跳的安全性。为适应集群系统不间断运行的特点，实现了鉴别密钥的在线替换。

关键词：机群 通信机制 计算机安全

集群是一种由软硬件相配合而构成高可用系统的解决方案。一组计算机互连起来并相互监控，若其中任一计算机发生故障，集群系统会将其应用和服务切换到其他计算机上运行。其中，心跳机制的任务是在集群结点之间传送集群系统的有关信息，如结点失败、结点恢复、加入或删除一个结点、在结点上启动或停止某项服务等。在由多个结点组成的集群系统中，心跳可通过广播或多点播送的方式传送。在收到心跳后，根据心跳附带中的信息，集群结点上运行的集群软件将采取对应的措施。

在集群软件的设计中，在强调对应用的保护和对系统资源检测功能的同时，心跳信息的安全性是易被忽略的一个方面。如果攻击者侵入网络，发送伪造的心跳信息，将导致集群软件采取错误的行动甚至造成灾难性的后果。因此，有必要在集群软件的心跳机制中加入安全性处理。

1 心跳的鉴别

结点之间传送心跳的介质可以是专有的，仅仅被集群结点所使用；也可以是共享传输介质（如以太网等）。为防止心跳传输成为单点故障，集群系统支持结点之间建立多条心跳传输路径，在某条心跳故障时仍能维持结点之间的通信。因此，除使用专有介质（如RS-232，NUMA等）以外，集群系统还使用共享介质作传输心跳。在共享介质中可能存在的窃听和欺骗同样对集群系统构成威胁。

在集群系统中，结点在发生故障时能够将应用切换到其他结点上，结点之间的心跳会导致集群软件执行敏感操作，例如，增加或删除结点的IP地址、重启动或关掉结点、挂接或卸载文件系统等。攻击者通过在心跳网络上伪造心跳信息，就有可能实现其目的。

在集群系统中，需确保只有事先指定的集群成员结点发送出的心跳才是有效的。因此，在心跳的传送中需要使用鉴别技术来验证心跳的合法性。

所有集群成员结点共享一个秘密密钥k，结点A在发送心跳信息m到结点B时，根据密钥k计算出信息m的鉴别码a，随心跳信息m发送给结点B。结点B在收到心跳信息m和鉴别码a后，根据与结点A相同的计算方法，根据密钥k计算出信息m的鉴别码a'，比较a和a'。如相等则证实心跳信息来自集群结点，否则说明心跳信息在传送过程中出现错误，或者是攻击者伪造的。鉴别的过程如图1所示。MAC为Message Authentication Code。

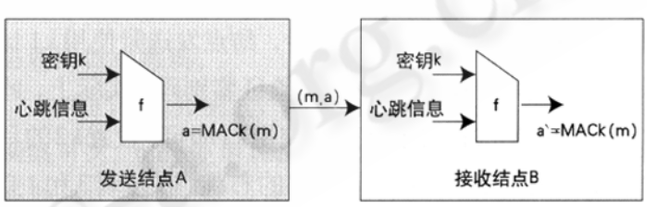

图 1 心跳信息的鉴别过程

在集群软件的实现中，采用了HMAC-SHA1算法。以保密散列算法SHA-1作为报文摘要函数，可以直接调用已发布的SHA-1库函数来计算鉴别码，不必在集群软件中重写此算法。HMAC的描述有如下的形式[1]：

$HMAC_k(x) = F(k \oplus opad, F(k \oplus ipad))$

其中F为保密散列算法，对于SHA-1，生成的摘要为160位。若k的位数不足160位，需加0补足。ipad和opad为160位常量，ipad规定为连续20个字节的0x36，opad规定为连续20个字节的0x5C。在调用SHA-1库函数的基础上，HMAC可自行实现，并用RFC2202提供的测试数据来验证其正确性。

MD5是另一个广泛使用的选择报文摘要函数，在心跳信息鉴别中选择使用SHA-1而不是MD5有两个理由。首先，SHA-1的摘要为160位，比MD5长32位，具有更好的安全性。其次，SHA-1的计算量开销比MD5大，但

由于心跳信息在结点之间的流量很小，不超过1KB/s，因此采用SHA-1并不会给系统带来很大负担[2]。

2 在线更新鉴别密钥

用于鉴别结点心跳信息的密钥由各结点共享，而定期更换密钥是安全策略的基本要求。在更新密钥时，一个简单的做法是停止集群软件，更新密钥到各个结点，再启动集群软件。这种做法违背了集群软件的设计初衷，集群软件运行在关键性场合，需不间断运行，甚至长达数年之久。因此，需设计一种机制，在集群软件运行时在线更换心跳鉴别密钥。

设原心跳鉴别密钥为k1，需更换为k2。分发密钥时指定一个密钥对（ka，kb），其中，ka为发送鉴别密钥，kb为接收鉴别密钥。发送心跳时使用ka计算鉴别码，接收心跳时使用kb验证鉴别码。更换过程中分为7个步骤：

(1) 将（k1, k1/k2）分发到所有结点

(2) 在所有结点上启用（k1, k1/k2）密钥对

(3) 将（k2, k1/k2）分发到所有结点

(4) 在所有结点上启用（k2, k1/k2）密钥对

(5) 将（k2, k2）分发到所有结点

(6) 在所有结点上启用（k2, k2）密钥对

在初始状态，所有结点在发送和鉴别心跳时使用鉴别密钥k1。在步骤1和2完成后，结点发送时使用k1计算鉴别码，接收时使用k1和k2验证鉴别码，实际上，此时收到的心跳信息的鉴别全部使用k1即可，设置k2的目的是能够识别步骤4后结点发出的心跳。在步骤3和4完成后，结点发送时使用k2计算鉴别码，接收时使用k1和k2验证鉴别码。此时，由于步骤2已经设置了k2为接收鉴别密钥，因此以k2为密钥鉴别码的心跳能够被结点所认可。步骤5和6执行完成后，在接收心跳时结点不再认可以k1为鉴别码的心跳，因为所有心跳发送时均使用k2为鉴别码。密钥使用的变化过程如表1所示。

表 1　密钥更新过程

时刻	发送鉴别密钥	心跳鉴别码	接收鉴别密钥
更新前	k1	HMACk1(m)	k1
步骤1、2执行后	k1	HMACk1(m)	k1 和 k2
步骤3、4执行后	k2	HMACk2(m)	k1 和 k2
更新完成后	k2	HMACk2(m)	k2

从上述步骤可知，在集群软件的心跳验证机制中，在更新密钥时需必须允许发送鉴别密钥和接收鉴别密钥不同，并且接收心跳后需使用新旧两个鉴别密钥来计算鉴别码。

除更换密钥外，还可用这种方法来更换鉴别过程中使用的散列保密算法，如将SHA-1替换为MD5。在分发密钥对时，需同时指出其散列保密算法。散列保密算法的实现置于动态连接库中，若引入原集群软件未实现的新的散列保密算法，还可替换其动态连接库。

3 抵御重发攻击

主动攻击者可记录心跳网络上传送的信息及其鉴别码，将其存储起来，在某一时机重发此心跳信息，导致集群结点执行攻击者希望的操作。因为鉴别码也随心跳信息一同出现在网络上，攻击者已获得该心跳的鉴别码，因此靠心跳鉴别的手段不能抵御重发攻击。

对付重发攻击的一个有效办法是在心跳信息上加时间戳，并指定其有效期。对超过有效期的心跳信息，可认为是攻击者的重发动作，予以丢弃。但这种方式要求结点之间的始终必须保持严格同步，在集群结点数量很多或由多种异构系统组成时，保持时间同步的要求是苛刻而不易实现的。

在考虑安全性问题之前，为避免收到网络中因为超时、拥塞等原因导致的心跳重发，心跳信息中已经包括了一个顺序号，每发送一次顺序号加1。结点重新启动后，顺序号从0重新开始。

顺序号的机制不能对付重发攻击，因为攻击者可记录心跳信息及其顺序号，在结点重新启动后，攻击者记录的心跳信息中所含的顺序号即被认可，通过重发这些心跳信息可导致集群软件采取错误动作。

因此，在心跳信息中，除顺序号外还需引入实例号。在结点重新启动后，在顺序号归0的同时，实例号加1。因此，心跳信息中的<实例号，顺序号>组合绝不会重复。与最近一次接受到的有效心跳<实例号，顺序号>相比较，攻击者所重发的心跳信息在任何时刻均能被识别出来。结点收到的心跳信息可分为以下几种情况：

表 2　重发攻击的判别

<实例号，顺序号>组合	解释	判定结果
新的顺序号，实例号不变	发送结点生成的新的心跳信息	心跳信息有效
新的实例号	发送结点重新启动后产生的心跳	心跳信息有效
旧的顺序号，实例号不变	以前曾发送过的心跳	重发攻击，丢弃
旧的实例号	以前曾发送过的心跳	重发攻击，丢弃

（下转第37页）

(上接第 43 页)

<实例号，顺序号>须作为心跳信息的一部分，作为报文摘要函数的输入来计算鉴别码，以避免攻击者伪造实例号和顺序号。

4 结论

在心跳机制中采用了HMAC-SHA1来鉴别心跳信息的有效性，并支持在线更新密钥以适合集群软件连续运行的要求。引如实例号可有效地抵御重发攻击。对于其他可能的攻击手段，如生日攻击、穷举攻击等，则由SHA-1算法本身及其最长达160位的密码来保证其安全性。

此外，可考虑自行实现新的保密散列算法，或将已有的保密散列算法SHA-1或MD5加以改变，对于无法阅读软件目标码的攻击者，其难度进一步增加。在密钥的分配方面，若指定每一个结点使用不同的密钥，可实现数字签名的功能，能追踪到来自结点内部的攻击。■

参考文献

1 M. Bellare, R. Canetti, and H. Krawczyk, "Message Authentication using Hash Functions — The HMAC Construction" [J] RSA Laboratories' CryptoBytes, 1996, 2(1):64–76。

2 王运凯，石磊，曹少文等. Java 2平台安全技术 [M]. 北京：机械工业出版社，2000.136–140。