SNMPv3用户安全机制的实现*)

The Implementation of the SNMPv3's User-based Security Mechanism

夏 云 邵军力

(解放军理工大学 南京210016)

Abstract This paper, based on the outlining SNMPv3's overall network management framework, analyzes the user-based security model, the procedure element of the security protocol and its implementation.

Keywords Security mechanism, User-based security model, Authenticaiton, Authority Agent engine, Encryption

1 引言

简单网络管理协议SNMP是应用最为广泛的网管协议,虽然SNMPv2在整体性能和部分功能上都较前版本有较大幅度的改进,但在其最后的标准版本SNMPv2c中仍然缺乏有效的安全特性[1],从而限制了该协议在政府、商业或军事等网络领域上的广泛应用。随着Internet的快速发展,IETF于1998年1月正式颁布了SNMPv3协议标准文档RFC2271-2275[2~6](事实上,目前最新的SNMPv3是RFC2570-2574),提出了网络管理中全面和系统化的安全解决方案,弥补了先前版本的不足。就安全特性而言,SNMPv3[8,9~11]是在SNMPv2u[7]提出的基于用户的安全模型基础上,经过深入评议、广泛意见征求和大幅修订后产生的,特别是完善了SNMPv2u的安全模型。本文将对SNMPv3的安全整体框架进行详细分析,重点阐述其基于用户安全模型的概念、协议安全处理的操作和设计实现过程。

2 网络管理协议实体组成结构

所谓网络管理协议实体是指SNMP标准所定义的SNMP框架的一个具体实现,SNMPv3的协议实体由一个SNMP引擎和一个或多个相关的SNMP应用所组成。SNMP引擎提供发送、接收、认证、加密消息和对被管对象的访问控制等服务,共包括四部分内容:①分发器:实现消息的发送和接收,确定消息版本号并将该消息发送给相应的消息处理模块,为SNMP应用提供抽象服务原语接口;②消息处理子系统:实现消息的发送准备和从接受到的消息中读取数据。它由若干个模块组成,不同模块实现对不同版本的消息处理;③安全子系统:实现消息认证和加密等服务,可以支持多个安全模型,而不同模型由相应模块来具体实现。SNMPv3采用基于用户的安全模型;④访问控制子系统:实现授权服务。可能支持多个访问控制模型,SNMPv3采用了基于视图的访问控制模型。

SNMP协议实体包含下列主要应用,它们使用了SNMP引擎所提供的各种服务:①命令产生器:监控和操纵管理数据(通常在管理者方实现);②命令响应器:实现对管理数据的访问(通常在设备代理方实现),③通告产生器:发送异步消息(如Inform、Trap等);④通告接收器:处理异步消息(通常在管理者方实现);⑤代理转发器:实现向不支持SNMP的设备转发消息(通常在设备代理方实现)。

为了保持各子系统间的独立性,SNMPv3在它们之间又定义了抽象服务接口。这些接口又由一系列原语组成,它们定义了原语在调用子系统服务时需要提供的抽象数据元素和返回的信息规范。因此,只要子系统的接口符合原语定义,具体的内部实现都可被更新或替换(如采用新的安全协议等),从而增加了协议的灵活性和可扩展性。

3 SNMPv3的安全威胁和安全模型的建立目标

在网管安全中,可能受到的威胁有以下六类[1],它们分别是:(1)信息篡改;(2)欺骗;(3)泄密;(4)消息流篡改;(5)服务否认;(6)业务流分析攻击。一般来说,一个网管系统的安全性需要针对以上威胁采取相应的安全机制和安全服务,而安全模型是实现这些目标的原则框架。

*)国家863重点研究课题:计算机网络管理与安全系统(1999-2001.863-306-ZD08-01-2)。夏 云 副研究员,博士生,从事的研究领域是TCP/IP网络协议、宽带IP技术和路由体系结构、网络管理与安全协议等。邵军力 教授,博士生导师,目前的主要研究领域是网络管理和安全技术等。

一个SNMP安全模型需要定义如下规范：防范的安全威胁、安全服务的目标以及提供认证、加密等具体安全服务的安全协议，而安全协议则定义了提供安全服务（如认证、加密等）的机理、过程和相关支持MIB数据的操作。

SNMP首先需要防范的是消息篡改和身份欺骗威胁，其次是泄密和业务流篡改威胁。因此，基于用户的安全模型需要实现以下目标：①验证所接收的SNMP消息的完整性，确保其在传输过程中没被篡改，防范消息篡改；②验证源发送者的身份，防范身份欺骗；③根据消息中的生成时间，保证消息从发送到接收之间的延迟在给定的时间窗口内，防范业务流篡改；④在需要的时候提供加密，确保消息的内容不被泄露，防范泄密威胁。

SNMPv3基于用户的安全模型提供如下服务：数据完整性服务、数据源端认证服务、数据加密服务和消息流（业务流）篡改攻击防护服务。在该模型中，数据源端认证和数据完整性服务是必须同时实现的，也就是说，两者或同时实现或都不实现；另外，数据机密性是建立在数据完整性和数据源端认证的基础之上，只有在实现完整性和数据源端认证之后才有可能提供机密性服务。

SNMPv3的安全模型由三部分组成：认证模块、时标模块和加密模块：①认证模块实现数据完整性和数据源身份认证，在RFC2274中规定必须支持HMAC-MD5-96[12]认证协议，同时也应该支持HMAC-SHA-96[12]认证协议；②时标模块实现防范消息延迟或重放攻击；③加密模块则保证消息内容不泄露，RFC2274规定必须支持CBC-DES加密协议。

4 基于用户的安全模型原理

4.1 用户属性

当SNMP引擎之间通信时，一方必须以在另一方上是合法用户的身份才能发送或接收消息。因此，每个SNMP引擎都在其本地数据库中维护了它所管用户的各类信息。引擎必须为每一个合法访问的用户维护下列属性参数：

用户属性项（变量名）	说　明
UseName	标识用户名的一个字符串
SecurityName	标识一个用户的字符串，其格式独立于安全模型
AuthProtocol	标识使用哪一个认证协议：HMAC-MD5-96或HMAC-SHA-96
AuthKey	若该用户发送的消息需要认证，则该参数指定认证协议使用的认证密钥
AuthKeyChange 或 AuthOwnKeyChange	定义了独立的远程安全更新认证密钥的方法
PrivProtocol	指示该用户发送的消息是否需要加密，若需要，采用何种加密协议。协议规定必须至少支持CBC-DES加密协议
PrivKey	若用户发送的消息要被加密，则该参数表示使用的加密密钥
PrivKeyChange 或 PrivOwnKeyChange	定义独立的安全远程更新认证密钥的方法

4.2 SNMPv3消息格式[2]

SNMPv3的协议消息格式如图1所示。其中消息头部分定义消息版本和消息管理参数，如：消息标识符、响应消息最大缓冲窗口、消息采用的安全模型以及该消息是否需要认证或安全检验标志等等。消息处理子系统则根据消息版本号将消息传递给处理不同版本消

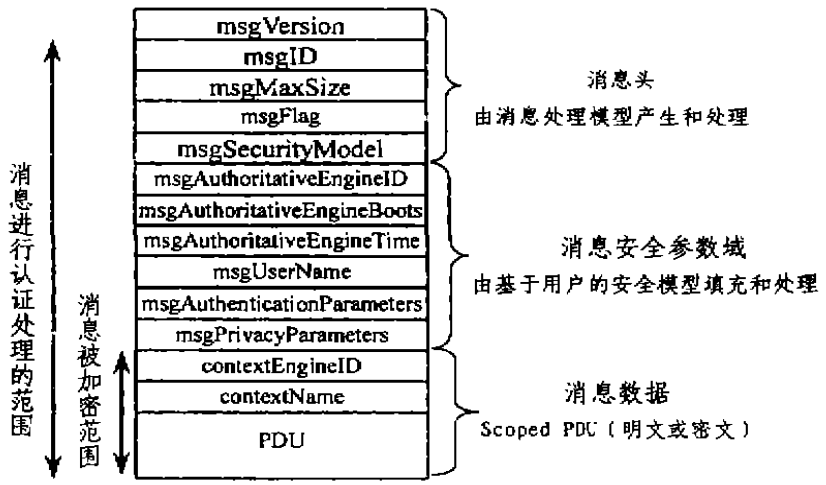

图1　SNMPv3消息格式

息的子模块进行处理；消息数据（scopedPDU）包含PDU和相关的访问控制信息。其中PDU是协议数据，而contextEngineID和contextName是与基于视图访问控制相关的二个字段（消息安全参数域的具体含义在下文说明）。

4.3 消息重放、延迟和重定向攻击的防范策略

为了防范消息重放、延迟和重定向，SNMPv3定义了SNMP权威引擎的概念，即参与通信的两个SNMP引擎中的一个必须充当另一个的权威引擎。

当一个SNMP操作（如Get，GetNext，GetBulk，Set或Inform）产生需要响应的消息时，则接收该消息的SNMP引擎是权威引擎。当SNMP消息不需要产生响应消息（如SNMPv2-Trap，Response或Report）时，则发送该消息的引擎是权威引擎。消息中安全参数部分中的消息权威引擎标识（msgAuthoritativeEngineID）字段标识通信中的权威引擎。每一个SNMP引擎都维护有三个对象，它们唯一地标识了该SNMP引擎的当前状态：①snmpEngineID：唯一地标识该SNMP引擎的标识符；②snmpEngineBoots：从该引擎ID被设置起，该SNMP引擎重新启动的次数；③snmpEngineTime：上次重新启动以后到现在已经运行的时间（秒）数。

每个引擎维护有自身的snmpEngineBoot和snmpEngineTime值作为引擎的时钟。每当引擎重新启动时，都要将其snmpEngineBoot增1，而将snmpEngineTime置为0。被次安装时的snmpEngineBoot和snmpEngineTime均为0。若snmpEngineTime到达最大值（$2^{31}-1=2147483647$），则snmpEngineBoot增1，尔后重置snmpEngineTime为0。

每个消息中都包含有通信中的权威引擎标识，即msgAuthoritativeEngineID字段。当接收到一个消息时，权威引擎检测该字段值以确保它是否是所希望的接收者。非权威引擎使用了接收消息中的该字段值来保证消息使用了正确的状态信息（非权威引擎在发送消息时，要有一份发出消息的状态信息备份，用来处理接收到的响应消息）并进行消息处理。

每个发送给权威引擎的消息或从权威引擎接收到的消息都要包含有标识本消息发送时刻的权威引擎的时钟值，它们必须由参数msgAuthoritativeEngineBoot和msgAuthoritativeEngineTime来共同指定。所以，一个非权威引擎在首次与某个权威引擎进行可信通信前，其首要任务是与该权威引擎取得时钟同步，并在每次SNMP权威引擎重新启动后，任何一个维护有该权威引擎的这两个值的引擎都必须首先与权威引擎重新取得时钟同步以继续进行两者之间的可信通信。在接收到消息以后，则要检测该值以确保消息的传输延迟是否在一定的时间窗口内（该模型规定时间窗口大小为150秒），这样可防范重放攻击的安全危险（非权威引擎还要维持另一个本地参数latestReceivedEngineTime以记录接收到的权威引擎发出的最近消息的时钟值）。

msgAuthenticationParameter参数字段存放消息认证的相关信息，如消息摘要认证码等内容，而msgPrivacyParameters参数字段存放消息加密的相关信息，如CBC-DES加密算法的初始化向量等内容（在消息接收端，这两个参数用来实现消息通信中的认证和加密）。

4.4 安全模型提供的服务原语[7]

1）消息接收服务原语

```
statusInformation=                          ——处理结果指示，成功或失败
  processIncomingMsg(
  IN messageProcessingModel                 ——SNMP版本号
  IN maxMessageSize                         ——消息源发送引擎可以接收的最大消息
  IN securityParameters                     ——安全参数
  IN securityModel                          ——对应于该安全模型
  IN securityLevel                          ——消息安全级别（仅认证或认证和加密）
  IN wholeMsg                               ——接收到的整个消息
  IN wholeMsgLength                         ——整个消息长度
  OUT securityEngineID                      ——权威SNMP引擎标识
  OUT securityName                          ——发送消息的用户标识
  OUT scopedPDU                             ——消息体（加密数据或非加密数据）
  OUT maxSizeResponseScopedPDU              ——响应消息PDU的最大尺寸
  OUT securityStateReference                ——安全状态参数指针，安全参数在发送响应消息
  )                                         ——时使用
```

2）消息发送服务原语

a）产生一个请求消息的服务原语：

```
statusInformation =
generateRequestMsg(
    IN    messageProcessingModel
    IN    globalData
            ——消息头（管理数据信息）
    IN    maxMessageSize
```

b）产生一个响应消息的服务原语：

```
statusInformation=
generateRequestMsg(
  IN    messageProcessingModel
  IN    globalData
  IN    maxMessageSize
  IN    securityModel
```

```
IN    securityModel
IN    securityEngineID
IN    securityName
IN    securityLevel
IN    scopedPDU
OUT securityParameters
        ——各种安全相关参数
OUT wholeMsg
OUT wholeMsgLength)
```

```
IN    securityEngineID
IN    securityName
IN    securityLevel
IN    scopedPDU
IN    securityStateReference
OUT securityParameters
OUT wholeMsg
OUT wholeMsgLength
)
```

5 消息安全检验处理过程

消息安全检查分为发送消息检查和接收消息安全检查两部分，图2分别给出了它们各自的简化工作流程。在具体的程序设计和实现过程中，SNMP 消息发送处理过程需要8个步骤来完成，而处理 SNMP 消息接收需要12个步骤，以下是它们具体的处理过程描述。

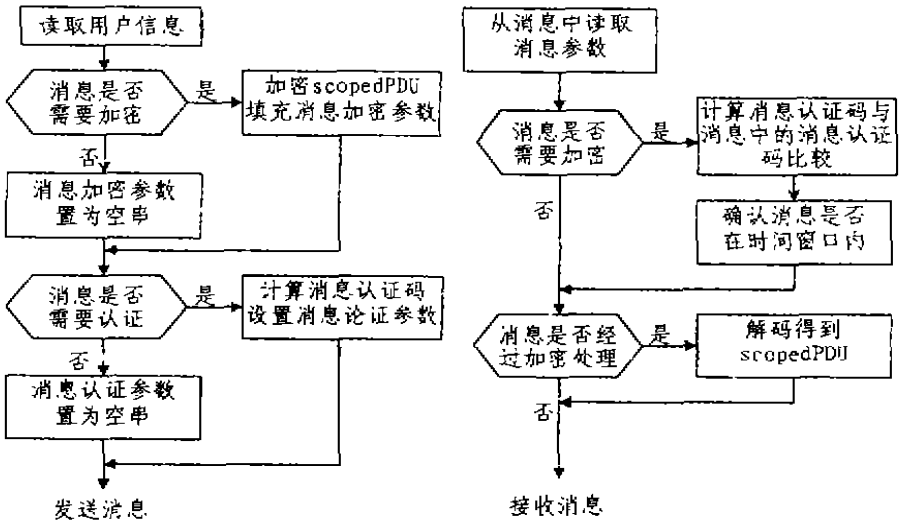

图2 安全相关的消息接受和发送的主流程

5.1 SNMP 消息发送处理过程

(1)若调用服务原语中产生的是响应消息原语，则从原语的安全状态指针所指向的安全参数状态中读出：用户相关信息、安全引擎 ID 和消息安全级别，然后将缓存的安全状态数据丢弃，否则(不是产生的响应消息)，则根据安全引擎 ID 从本地配置数据库(LCD)中读出参数指定的安全名所对应的用户相关信息，如无该用户的信息，则返回不可识别的安全名错误指示；

(2)若安全级别指定需要对消息进行加密或认证处理，而用户不提供相应的认证和加密支持，则返回不支持的安全级别错误指示；

(3)a) 若安全级别指定消息需要加密，则根据用户加密协议对 scopedPDU 使用下面的原语进行加密[5]：

```
statusInformation=
   ——操作成功与否指示信息
encryptData(
      IN encryptKey
         —— 用户的本地加密密钥
      IN dataToEncrypt
         —— 经过 ASN.1编码的 scopedPDU
      OUT encryptedData
         ——经过 ASN.1编码的加密 PDU
      OUT privParameters
         ——经过 ASN.1的加密参数信息
)
```

若处理出现错误，则消息不能被发送，返回错误指示；若成功，则将返回的加密参数信息填入到消息中的消息安全参数字段中；

b)若安全级别参数指定消息无须加密操作，则对 NULL 串进行 ASN.1编码，并将其填入到消息中的消息安全参数字段中；

(4)将经过 ASN.1编码的引擎 ID 消息中相应字段，对于请求消息来说，该字段可以为空，因为这将导致远程 SNMP 引擎返回一个 report 消息，该 report 消息的消息权威引擎 ID 字段包含接收消息者的引擎 ID；

(5)若原语的安全级别参数指示需要对消息进行认证处理，则使用安全引擎 ID(若是响应消息，则该值为本地引擎 ID)参数作为索引查询本地配置数据库，用得到的引擎重启次数和引擎时钟值用来作为消息相应字段值；

(6)将经过 ASN.1编码的用户名作为消息用户名

字段值；

(7)a)若安全级别指定需进行认证服务，则调用下面的原语进行认证服务[5]：

```
statusInformation=
authenticateOutgoingMsg(
        IN authKey
            ——用户本地认证密钥
        IN wholeMsg
            ——没有经过认证处理的消息
        OUT authenticatedWholeMsg
            ——经过认证处理得到的消息
)
```

若返回错误指示，则消息不可发送；若返回成功信息，则将相应的认证信息填入消息字段，对消息进行ASN.1编码；

b)若指示消息无须认证服务，则将经过ASN.1编码空串填入认证消息参数字段中，再对整个消息进行ASN.1编码；

(8)安全处理结束以后，将整个消息及其长度返回给调用模块。

5.2 SNMP消息接收处理过程

(1)当接收到的消息不是合法的ASN.1编码位串时，则snmpInASNParseErr计数器增一并返回解析错误指示；

(2)从消息中解析出安全参数字段值：用一个指针指向保存下来以备响应消息使用，这些安全参数是：msgUserName、securityEngineID和securityLevel；

(3)若安全参数中权威引擎ID字段的值不可知，则：a)若非权威引擎具有权威引擎发现功能，则其会在LCD中创建一个新的表项，然后继续处理；b)否则，usmStatsUnknownEngineID计数器加一，并返回不可识别的引擎ID错误指示，并将该计数器的OID/value返回给调用模块；

(4)从LCD中读取有关与该用户相对应的信息，若没有该用户信息，则usmStatsUnknownUserName计数器增一，返回不可知的安全名错误指示并将该计数器的OID/value返回给调用模块；

(5)若用户信息指示不支持消息中相关参数字段所要求的安全级别，则计数器usmStatsUnsupportedSecLevel增1，返回不支持的安全级别错误指示，并将该计数器的OID/value返回给调用模块；

(6)若安全级别指定消息使用了认证，则通过如下原语调用相应的支持用户认证协议的认证模块进行安全处理[5]：

```
statusInformation=
   ——成功或失败指示
authenticateIncomingMsg(
    IN authKey
        ——用户本地认证密钥
    IN authParameters
        ——认证信息
    IN wholeMsg
        ——整个消息
    OUT authenticatedWholeMsg
        ——经过认证处理的消息
)
```

若认证失败，则丢弃消息，usmStatsWrongDigest计数器增1，返回认证失败错误指示，并将该计数器的OID/value返回给调用模块；

(7)若该消息经过认证处理并认为是可信消息，则从LCD中读入snmpEngineBoot和snmpEngineTime值并进行以下处理：

a)若消息中的引擎ID与处理消息的引擎ID相同(消息接受者为一权威引擎)，则如果出现下面任何一种情况，则认为消息超出时间窗口：

—本地snmpEngineBoot为2147483647(达到最大值)

—消息中的权威引擎重启次数字段值不等于本地的snmpEngineBoot值

—消息中的权威引擎时间值与本地的时间值差异大于+/—150秒

若认为消息超出时间窗口，则usmStatsNotInTimeWindow计数器增一，返回不在时间窗口内错误指示及该计数器的OID/vlaue给调用模块。

b)若从消息中读取出来的权威引擎ID不等于处理消息的引擎ID(接收消息者为非权威引擎)，则：

i)若任何下面的条件为真时：

—消息中的权威引擎重启数值大于本地值，或

—重启次数值相等，而消息中的引擎时间大于latestReceivedEngineTime值，则对应于消息中的权威引擎ID的LCD中的相应表项被更新，实现时钟同步：

—引擎重新启动次数设为消息中相应字段的值

—引擎时钟值被设置为消息中的时钟记数值

—最近接收消息的时间变量被设定为消息中的时钟值

u)若任何一个下面的条件为真，则消息被认为超出了时间窗口：

—本地引擎重启次数记录等于2147483647

—消息中权威引擎启动次数字段值小于本地值

—重启次数相等，但是消息中记录的时钟值比本地时钟值小150秒以上。

若消息被认为是超出了时间窗口，则返回notInTimeWindow错误指示，但是该过程允许消息中的权威引擎重新启动次数大于本地引擎重启次数，即对这样的一类消息也被认为是可信消息。权威引擎再次重新启动时就会出现这种情况，这时，非权威引擎应该通过该消息来更新其保存的权威引擎的时标值，以重新取得时针同步。

(8)a)若安全级别指示消息经过加密处理，则通过下面的原语调用安全子系统中的加密模块对加密的

PDU 进行解密处理[5]：

```
statusInformation=
    ——处理结果指示成功或失败
decryptData(
        IN decryptKey
            ——用户加密密钥
        IN privParameters
            ——加密协议相关参数
        IN encryptedData
            ——加密数据
        OUT decryptedData
                ——解密以后的数据
)
```

若加密模块出现错误，则消息不能进一步处理，usmStatsDecryptionErrors 计数器增一，返回解密失败错误指示及该计数器的 OID/value 给调用模块；若处理成功，则将解密得到的数据返回给调用模块；

b)若安全级别指示消息没有经过加密处理，则直接将消息返回给调用者；

(9)计算响应消息允许的 PDU 最大长度；

(10)根据消息中的用户名字段从基于用户的安全模型的用户信息表中读取用户的安全名(securityName)；

(11)将与该消息相关的安全数据缓存下来，以供响应消息处理时使用。与对应的请求消息相同的安全参数，要缓存如下信息：msgUserName，usmUserAuthProtocol，usmUserAuthKey，usmUserPrivProtocol，usmUserPrivKeysecurity，EngineID，和 securityLevel；

(12)处理结束，返回调用子系统处理成功信息和消息中的 PDU 部分。

结束语 SNMPv1-2都采用了基于共同体名串的简单安全模型，其安全性能很差。虽然在 S-SNMP，SNMPv2P 和 SNMPv2u 分别提出了基于参加者(party)和基于用户的安全模型，但由于其操作复杂臃肿，与 SNMP 强调简单性的设计思想背道而驰，故没有在工业界得到实现和广泛支持。SNMPv3的提出既保证了安全性又保持了操作的简单性，弥补了 SNMP 版本的不足，可以预见它将很快成为业界流行的支持标准。本文的研究成果将指导我们完成863重点研究项目：计算机网络管理与安全系统。

参 考 文 献

1 夏云，等. SNMPv1-2协议的安全性分析 通信工程学院学报，1999，13(2)
2 RFC 2271-1998 An Architecture for Describing SNMP Management Frameworks
3 RFC 2272-1998 Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
4 RFC2273-1998 SNMPv3 Applications
5 RFC2274-1998 User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
6 RFC 2275 - View-based Access Control Model (VACM) for SNMPv3
7 RFC1910-1996 User-based Security Model for SNMPV2
8 Stalling W. SNMPV3. A Security Enhancement for SNMP. IEEE Communications Surveys，1998，1(1)
9 Stalling W. SNMP，SNMPv2，SNMPv3，and RMON 1 and 2. Third Edition. Prentice Hall PTR，1999
10 Zeltserman D. Brookline and Massachusetts Practical Guide to SNMPv3 and Network Management，First Edition. Prentice Hall PTR：1999
11 Stalling W. Security Comes to SNMP：The New SNMPv3 Proposed Internet Standards. Internet Protocol Journal，1998，1(3). Cisco News Publication Group，Cisco Systems
12 RFC 2104-1997 HMAC：Keyed-Hashing for Message Authentication

---

(上接第111页)

在运用 SSE-CMM 模型评估一个组织的过程能力成熟度之前，应首先使用这一模型评估该组织在以往工程项目中的表现。

由于每个能力级别都定义了一个或多个共同特性，只有当所有共同特性都得到满足时，才达到了对应的能力级别。如果一个过程域只满足了 n+1级或 n+2级上所定义的部分共同特性，但满足了 n 级上所定义的全部共同特性，其过程能力应当评为 n 级。

在执行具体项目时，一个组织可以根据系统安全工程项目的实际需求有选择地执行某些过程域，而不是全部。此外，一个组织也可能需要执行安全工程过程域之外的关键过程。SSE-CMM 模型推荐了 SSE-CMM 模型的11个过程域，它们可用于组织和项目本身的管理，可以与 SSE-CMM 过程域配合使用。

为了支持理论模型，保障过程能力评估结果的一致性，SSE-CMM 项目组编写了 SSE-CMM 模型评估方法指南。评估方法指南详细地规定了评估机构的组成、人员责任的区分、日程的安排、评估过程中所使用的一些表格格式及内容等。评估过程包括持续一周的与被评组织直接接触的调研活动。指南建议的评估时间是：自我评估为500人小时左右，第三方评估为大约1000人小时左右。评估活动本身并不复杂，主要是确认 SSE-CMM 模型中定义的基本实践和通用实践是否存在。被评组织必须提交证据以支持自己的论点。

参 考 文 献

1 SSE-CMM Model Description Document Version 2.0. April 16，1999
2 SSE-CMM Appraisal Method Version 2.0. April 16，1999
3 http:\\www.sse-cmm.org.
4 http:\\www.sse-cmm.org/library/