目录

[基于SVM的Modbus TCP深度包检测 1](#_Toc495996354)

[1.1 深度包检测技术 1](#_Toc495996355)

[1.2 Modbus TCP深度包解析 2](#_Toc495996356)

[1.3 Modbus TCP深度包过滤 6](#_Toc495996357)

[1.4 基于SVM的异常检测 9](#_Toc495996358)

基于SVM的Modbus TCP深度包检测

针对Modbus TCP协议的安全缺陷，将使用深度包检测技术对协议数据进行数据链路层到应用层的全面解析，基于白名单的方式对数据包内的各种关键信息进行过滤。进一步的深度检测使用异常检测方法，选取功能码和线圈或寄存器的起始地址作为特征，基于SVM（支持向量机算法）建立异常检测模型，识别非正常流量。

深度包检测技术

深度包检测技术简称DPI，它是一种能够对应用层的数据进行流量检测和控制的技术。当TCP/IP数据流通过DPI系统时，该技术可以深入读取IP数据包的载荷内容，并且对OSI模型中的应用层信息进行重组，从而获得数据流的内容，然后按照定义的规则策略对数据进行过滤操作[34]。

DPI的“深度”是相较于普通报文检测而言的，传统的检测方法仅能够对OSI模型中的网络层和传输层的内容进行检测。由于这两层数据的组织结构是相对固定的，如表1-1、表1-2所示为网络层的IP数据包结构和传输层的TCP数据包结构，其中TCP数据包为IP数据包结构中的用户数据部分。IP数据包头部中包含32位的源地址和32位的目的地址，TCP数据包头部包含16位的源端口和16位的目的端口，所以传统检测方法能检测的内容只包括源IP地址、目的IP地址、源端口、目的端口。DPI技术除了对上述内容进行检测识别外，还增加了对应用层内容的分析识别。

表1-1 IP数据包结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 头长 | 服务类型 | 包裹总长 | |
| 重组标识 | | | 标识 | 段偏移量 |
| 生存时间 | | 协议代码 | 头校验和 | |
| 32位源地址 | | | | |
| 32位目的地址 | | | | |
| 可选选项 | | | | |
| 用户数据 | | | | |

表1-2 TCP数据包结构

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 源端口 | | | | | | | | 目的端口 | |
| 数据序号 | | | | | | | | | |
| 确认序号 | | | | | | | | | |
| 偏移 | 保留 | U | A | P | R | S | F | 窗口字段 | |
| 包校验和 | | | | | | | | 紧急字段 | |
| 可选选项 | | | | | | | | | 填充 |
| 用户数据 | | | | | | | | | |

深度包检测技术实现了对数据包更深层次的检测，不仅能够检测单个数据包的内容，还能够把分散的数据包重新组合成相关联的数据流，在保持数据流状态的同时进行攻击的检测。

传统的检测技术由于检测内容有限，是一种快速的检测方式，但是目前大多数的攻击都是针对应用层的攻击，恶意的代码或者指令往往隐藏在应用层内容之中，透过缺乏应用层检测的防火墙，对内部网络发动攻击。传统的检测技术缺乏对应用层内容的识别能力，已经无法满足现阶段的安全需求。与传统的检测技术相比，DPI的优势主要包括：更深层次的数据检测；更低的攻击漏检率；更强的防护能力。

深度包检测技术实现对网络数据包的深度检测，对抵御网络攻击起到很好的作用。深度包检测采用了多种检测攻击的方法，包括协议分析、异常流量检测和特征模式匹配。

Modbus TCP深度包解析

Modbus TCP协议依靠面向连接的TCP技术实现了高速稳定的传输，它拥有较为固定的协议格式，表1-3显示了应用层Modbus TCP协议的具体格式以及每部分所占的字节数。

TCP/IP协议是一个具有鲜明层次的模型，每一层都具有严格标准的封包规范。数据包从应用层、运输层、网络层、数据链路层依次按照标准规范一层层的进行封包，具有高度的有序性。深度包检测技术中的协议分析正是利用了网络数据包具有高度的有序性和相关协议具体内容，结合高速数据包捕获、命令解析来进行检测，判别是否存在攻击。

表1-3 Modbus TCP数据结构

|  |  |  |
| --- | --- | --- |
| 事务处理标识符  （2字节） | 协议标识符  （2字节） | |
| 长度  （2字节） | 单元标识符  （1字节） | 功能码  （1字节） |
| 数据  （n字节） | | |

协议分析技术将收集到的网络数据包当成具有固定格式的数据流。首先它按照网络协议报文封装的反顺序对这些数据包进行解封。然后把各层协议数据内容与网络协议的标准（RFC 内容）进行逐层分析。协议标准都规定了数据报文中应该具有的协议字段以及期望值。协议分析技术的工作就是把当前数据包中各层协议字段的内容与协议定义的期望值进行比较，检查其是否符合协议的规定。如果符合相关协议的规定，就认为是合法的数据包。如果当前数据包的协议字段不符合协议的规定，则认为当前的数据包为非法网络流量，执行丢弃数据包的处理。

为了切实保护工业控制网络中Modbus TCP协议的信息安全，需要对Modbus TCP报文进行深度解析，发现攻击与入侵行为。对其进行深度包检测一般采取协议分析技术，逐层进行解析。图1-1给出了Modbus TCP深度解析的一般模型。



图1-1 Modbus TCP深度解析模型

该模型主要包括4个部分：数据链路层检测、网络层检测、传输层检测和Modbus TCP应用层检测，其中应用层检测又可分为MBAP报文头检测和报文内容检测。主要检测过程如下：

1 数据链路层检测：该部分主要解析数据包中的源mac地址与目的mac地址。mac地址是网卡的物理地址，一个网卡拥有全球唯一的固定的mac地址。对mac地址进行过滤可以阻止非法的硬件设备对Modbus TCP资源的访问。

2 网络层检测：该部分主要是解析数据包中网络层的源IP地址和目的IP地址。IP地址又称网际协议地址，它是网络为每一台主机分配的逻辑地址，用来屏蔽物理地址的差异，IP地址是可变的。对IP地址进行检测可以获得访问者的身份信息，为访问控制和入侵检测提供支持。

3 传输层检测：在传输层主要进行源端口号和目的端口号的检测。不同的应用一般使用不同的端口进行通信。Modbus TCP使用502端口进行通信，对端口进行过滤可以阻止非法的应用对Modbus资源的连接与访问。

4 应用层检测：本部分是Modbus TCP深度包检测的主体，包括对协议MBAP头部信息的检测和数据部分的检测。

4.1对MBAP头部信息的解析包括事务处理标识符、协议标识符、长度和单元标识符。其中，Modbus TCP的协议标识符固定为0x0000，对协议标识符进行检测可以判定是否是Modbus通信；长度代表了之后数据的总长度，即单元标识符、功能和数据的总长度，对长度进行检测可以判断出是否是恶意构造的数据包；单元标识符用来指明和主设备相连接的从设备的识别码，对此单元标识符进行检测可以对不同的角色进行权限控制，阻止一些用户访问非授权的设备。

4.2对Modbus TCP协议数据部分的检测主要包括功能码以及寄存器和线圈地址的检测，在必要时还可对写入寄存器的数据进行检测，该部分是深度包检测的核心与关键。

功能码代表了主机对从设备的控制和操作意图，对功能码进行严格的过滤，阻止非法的控制请求，保护Modbus设备避免造成严重的生产事故。

寄存器和线圈地址以及之后的数据一般与功能码相对应，Modbus规定了4种基本数据类型，分别为离散量输入、线圈、输入寄存器和保持寄存器，他们分别应用于特定的功能码。例如，功能码02表示读输入离散量，操作对象为占1比特的离散量输入；功能码05表示写单个线圈，操作对象为线圈；功能码04表示读输入寄存器；功能码06表示写单个寄存器，操作对象为保持寄存器。对协议报文数据部分地址和数值的检测需要结合功能码来进行。

图1-2给出了Modbus数据部分检测的模型，检测过程主要分为以下几部分：

（1）对功能码进行检测，如果和4种基本数据类型的操作无关，则执行步骤（2），否则转到步骤（3）；

（2）检测是否存在功能码子码，若存在则根据相应功能码与子码的关键字段进行检测，然后转到步骤（7）；

（3）判断操作的数据类型；

（4）检测离散量、线圈或寄存器的起始地址。

（5）检测操作的数量。

（6）检测写入对象的数值（若果有写入操作，若无则跳过该步骤）；

（7）流程结束，获得报文的关键信息。



图1-2 Modbus数据检测

通过Modbus TCP深度解析，我们可以得到数据链路层、网络层、传输层、应用Modbus的关键信息，表1-3显示了深度解析所获取的各层信息。通过所获取的信息，可以得知是谁（mac地址和IP地址）发出了Modbus指令，并且知道进行怎样的操作（功能码），以及操作的对象（起始地址）。对所获取的信息分析，可以初步识别出非法的数据包，例如协议标识符不符合要求或者长度不匹配等，同时，利用所获信息可以进行更深一步的检测与分析。

表1-3 深度解析信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据链路层 | 源mac地址 | | 目的mac地址 | |
| 网络层 | 源IP地址 | | 目的IP地址 | |
| 传输层 | 源端口 | | 目的端口 | |
| 应用层  modbus | 协议标识符 | 长度 | | 单元标识符 |
| 功能码 | | 起始地址 | |

Modbus TCP深度包过滤

通过上述Modbus TCP深度包解析方法，我们可以得到从数据链路层到应用的关键信息，从而为深度数据包过滤策略的制定提供了支持。通过深度包过滤去除掉明显的非法数据包，可以使被保护网络的安全性大幅提高，同时对于隐蔽性更高的欺骗性等攻击，深度包过滤为进一步的异常检测提供了预警信息和过滤基础，使得异常检测更加高效。

深度包过滤一般采用黑名单和白名单两种方式：

黑名单模式是指总结网络中存在的不同攻击的特征，将这些攻击特征使用具体的规则进行描述，创建包含攻击特征的黑名单。在具体过滤时，将网络数据包与黑名单内的规则进行一一比对，如果数据包与某条规则相匹配，则认为该数据包存在攻击行为，拒绝该数据包的通过。如果数据包与黑名单内的规则均不匹配，则认为数据包合法，那么就会执行默认的过滤策略，即允许数据包通过。黑名单模式一般用于通信协议较多且繁杂的传统IT网络。

白名单模式是指将允许通过的数据包总结其具有的特征，将这些特征使用具体规则描述，创建允许通过的特征白名单。在过滤时，将数据包与白名单内的规则进行匹配，如果与其中的某一条项吻合，则认为是需要通过的数据包，让其正常通行。如果与白名单内的规则都不匹配，则执行默认的拒绝策略，不允许该数据包通过。白名单一般用于通信协议比较单一的网络。

安全规则的描述由于技术和设备的不同而有多种多样的表现形式，下面给出一种规则的描述形式：

［Action］ ［mac］ ［IP］ ［Port］ ［Protocol］

其中，Action表示对后面各项内容相匹配时所采取的动作，包含3中情况：

Allow：允许数据包通过；

Deny：拒绝数据包通过并进行报警；

Alert：对该数据包进行报警。

[mac]选项包含源mac地址和目的mac地址；[IP]选项包括源IP地址和目的IP地址；[Port]选项包括源端口和目的端口；[Protocol]选项表示对应用层协议部分的匹配。针对Modbus TCP协议，可以将该部分设置为：

［PID］［UID］［FC］［SA］［Num］［Value］

其中，PID为MBAP报文头中的协议标识符，UID为单元标识符，FC表示功能码，SA为操作对象的起始地址，Num为操作对象的数量，Value为操作对象的数值。在部分Modbus指令中，SA、Num、Value可能不存在，具体情况可根据实际进行设置。

根据上述的规则描述形式（部分选项可以为空），即可设置过滤策略。对于工控网络的Modbus TCP协议，由于其网络通信较为单一，并不存在IT网络那种繁杂的多协议通信，故一般采用白名单模式进行安全防护，仅允许白名单内的IP地址、端口、功能码、起始地址等通过，阻止一切白名单之外的通信流量通过。基于白名单的安全防护方法如图1-3所示。



图1-3 深度包过滤规则设置方法

1、设置Modbus TCP的白名单

白名单包括mac地址、IP地址、端口号、单元标识符、功能码、线圈或寄存器地址范围。

对于白名单有两种设置方式：

表1-4 白名单设置示例

|  |  |
| --- | --- |
| 白名单选项 | 内容 |
| 源IP | 172.16.10.1,172.16.10.2，….，172.16.10.10 |
| 目的IP | 172.16.10.99 |
| 源端口 | >10000 |
| 目的端口 | 502 |
| 单元标识符 | 1,2,3,… |
| 功能码 | 1-8,11-17,… |
| 线圈或寄存器地址 | 0x0000-0x1111 |

（1）简单白名单模式：只需对上述白名单选项进行单独设置，如表1-4所示，列出了白名单的具体内容。规则设置如下所示：

［Action：Allow］［IP:源IP列表->目的IP列表］［Port：源端口列表->502］［UID：单元标识符列表］［FC：功能码列表］［SA：地址列表］

（2）复杂白名单模式：根据实际需求对白名单内的各种选项进行组合设置，例如，只允许172.16.10.1这一IP地址的3524端口访问Modbus服务器，并且只能使用06功能码在0x0000地址写入0x6666这个值，则规则设置如下：

［Action：Allow］［IP:172.168.1.1->172.168.1.99］［Port：3524->502］ ［FC：06］［SA：0x0000］［Value：0x6666］

简单白名单模式的设置方便简洁，不需要对Modbus协议有深入的了解即可进行配置，此方法可以过滤掉绝大部分的非法数据包，但是对于一些恶意构造的数据包却不能进行有效的防护，比如，一个构造的数据包中功能码和线圈地址都在白名单内，但该线圈地址并非该功能码的有效操作对象。若采用复杂白名单模式，为了保证工控网络中Modbus TCP通信的正常进行，则需要设置大量的白名单规则，且需要对协议特征、控制流程等方面要有深度的了解方可，规则的修改也比较麻烦，但是却能对Modbus的通信提供更深入的防护。实际应用中通常采用普通白名单模式。

2、设置报警规则

对于一些无法完全认定为合法的操作进行报警处理。由于Modbus中的部分通信功能可能被攻击者利用，但系统本身也要用到，故此对这部分操作进行报警而不是拒绝其通过。比如08功能码的01子码为重启设备命令，该操作要慎重对待，故此进行报警处理。规则示例如下：

［Action：Alert］［IP:172.168.1.1->172.168.1.99］［Port：any->502］［UID：01］［FC：08］［SubFC：01］

3、设置默认拒绝规则

对于没有通过1和2中所设置规则的Modbus TCP数据包，都设置为默认拒绝，阻止该数据包的通过并进行报警。规则示例如下：

［Action：Deny］［mac：any->any］［IP:any->any］［Port：any->any］［UID：any］［FC：any］

基于SVM的异常检测

基于白名单的安全防护方法对已知的攻击入侵以及Modbus TCP的协议缺陷等问题提供了有效的解决方案，但是该方法难以描述同时存在于若干个数据包中的通信特征，对于上述的不足以及可能存在的新型攻击进行识别防护，首先通过深度包过滤去除掉不符合自定义规则的数据，然后通过异常检测对过滤后的数据进一步检测，识别不符合正常通信规律的数据流量，发现隐藏在其中的攻击行为。

对采用Modbus TCP协议的工业控制系统通信进行异常检测，之前研究者一般通过对Modbus TCP协议功能码的异常检测来实现：通过对获取的主设备发送给从设备大量的数据包中功能码的统计分析，实现对通信的异常检测。

功能码作为Modbus TCP报文部分的关键内容，每个特定的功能码代表不同的控制执行操作，可以有效的表明客户端对服务器端的操作意图，具有重要的研究意义。

线圈或寄存器地址作为硬件设备的物理地址，对于设备的安全性保密性至关重要，并且线圈或寄存器地址与功能码有着很强的对应关系，例如01功能码为读线圈，其后所跟的地址只能是线圈地址。然而对线圈或寄存器地址的异常检测研究却少之又少，同时考虑到功能和线圈地址，并将其对应起来进行异常检测更是空白。故此，选取功能码和线圈/寄存器的起始地址作为研究对象，对其进行异常检测。

在Modbus协议规范中，正常且具有实际意义的功能码只有19个，其中比较常见的主要是取值1、2、3这3个读操作功能码和取值5表示写操作的功能码。寄存器/线圈包含在数据部分，其中包括了起始地址和参考号两部分，起始地址表征将被查询的slave端的起始地址，参考号表征slave端被查询的数量。



图1-4 Modbus通信过程

图1-4简单表示了master端和slave端的通信过程。由于这里只表征modbus的请求与响应过程，只标出数据包中的功能码和相关数据部分。在图中，master端发出请求包，包中包含了功能码（FC），寄存器/线圈起始地址（SA），参考号（WC）。当slave端收到请求包后，向master发回一个响应包，响应包中包含了功能码（FC），字节计数（BC）以及输入状态或者设置内容等。

工业控制系统中，控制过程与事件的执行具有明显的周期特性。Modbus TCP 工业网络在稳定运行时所产生的数据流量也具有一定的序列特征和行为模式。而遭病毒感染的master端或入侵者向slave端发起攻击时，所产生的 Modbus 报文肯定会破坏Modbus TCP通信流量稳定的行为特征。所以对Modbus TCP控制网络正常运行时的通信流量建立网络通信行为模型，即可识别出异常的通讯或者攻击入侵行为。选择功能码和线圈/寄存器起始地址作为特征，将连续的modbus通信数据包转变为抽象化的modbus功能码和起始地址组合对序列，即是将Modbus TCP网络通信异常检测转化为modbus功能码和起始地址的异常检测。