doi:10.3969/j.issn.1671-637X.2016.03.012

面向SDN的源地址验证方法研究

孙 鹏
(中国电子科技集团公司电子科学研究院,北京 100041)

摘 要:当前互联网上出现越来越多的基于源地址欺骗的网络攻击,这类攻击很难被追查,对网络安全造成巨大威胁。在传统网络条件的限制下,实现源地址验证会遇到很多困难。得益于软件定义网络(SDN)带来的网络革新,网络控制变得更加便捷。面向SDN架构,利用可编程控制器对源地址验证方法进行重新设计和实现,提出两种面向SDN的源地址验证方法:一种是将无状态的IP地址与底层不可变标记如MAC地址、端口号绑定起来,在交换机中形成(MAC地址,端口号,源IP地址)三元组流表的过滤规则;另一种是利用最短路径算法计算路由路径,向路径上交换机下发(源IP地址,目的IP地址,入端口,出端口)四元组流表作为过滤准则。最后进行仿真实验,比较两种方案的实验结果。

关键词:网络安全;软件定义网络;源地址验证
中图分类号:TP393.02 **文献标志码**:A **文章编号**:1671-637X(2016)03-0049-05

Source Address Validation Methods Based on SDN

SUN Peng
(China Academy of Electronics and Information Technology, Beijing 100041, China)

Abstract: Nowadays, more and more attacks based on source address spoofing appear on the internet, which is difficult to trace and is a big threat to network security. Under the condition of the existing network environment, it is very difficult to implement source address validation. A significant network innovation brought by Software-Defined Networking (SDN) has made the network control more convenient. This article utilizes programmable controller to redesign and implement source address validation method, and puts forward two kinds of source address validation methods based on SDN. One is binding the stateless IP address and underlying immutable tags like MAC address/Port, forming a triple flow table filtering rules (MAC, Port and IP) in the interchanger; the other is to compute routing path with the shortest path algorithm, sending flow tables like source_IP, destination_IP, in_port and out_port as filtering rules. Simulation experiment was made to compare the effect of the two schemes.

Key words: network security; Software-Defined Networking (SDN); source address validation

0 引言

软件定义网络(Software-Defined Networking,SDN)诞生于美国斯坦福大学的Clean Slate项目,以MICKEOWN N教授为首的团队提出OpenFlow概念[1]用于校园网的试验创新,将传统的控制平面和转发平面相分离,以网络操作系统的形式和自由可编程的方法统一管理网络设备,提高网络控制灵活性、开放性,符合未来互联网的发展需求,使其一经提出便成为学术界和产业界的热点概念。

传统网络转发基于TCP/IP协议栈,源主机将自己的源地址填入报文首部,路由器只根据目的地址进行转发而不对源地址的真实性做出检查。这使得攻击者很容易假冒他人身份,向报文中填入虚假的源地址,发动诸如DDoS攻击、IP地址欺骗、TCP SYN flood攻击等网络攻击,源地址验证技术便是解决这个问题的方法之一。

在此背景下,研究SDN工作机制,将SDN与现有技术结合是发展SDN的一个重要方向,将源地址验证技术在SDN上实现,对于SDN的推广和网络安全具有积极意义。

收稿日期:2015-03-24 **修回日期**:2015-04-20
基金项目:"十二五"装备预研项目(41001010102)
作者简介:孙 鹏(1990—),男,四川绵阳人,硕士,研究方向为网络安全、软件定义网络。

1 源地址验证相关技术

目前学术界已提出多种源地址验证技术，依据设计方法和工作思路的不同，主要分为源地址端到端验证机制、源地址路径过滤验证机制和事后追踪机制。

1.1 源地址端到端验证机制

此类验证机制主要是在端节点处对源地址进行验证工作，使得报文接收端在接收报文时就能验证源地址的真实性，可以通过在报文的发送端添加签名，在接收端根据签名验证报文源地址真实性。典型的代表技术有 IPSec，APPA[2] 和 SPM[3] 等。

1.2 源地址路径过滤验证机制

此类验证机制主要是在数据包转发路径上依据不同的准则进行验证工作，转发路径上的中间节点如路由器、交换机等具有源地址验证、过滤的功能，在接收端未受到伤害时将攻击报文清除。典型的代表方法有 Ingress Filter[4]，uRPF[5]（unicast Reverse Path Forwarding），DPF[6]（Route-Based Packet Filter），HCF[7]（Hop Counter Filter）和 SAVA[8]（Source Address Validation Architecture）等。

1.3 事后追踪机制

此类机制无法保护终端不受网络攻击，但在攻击后，可以对攻击包的源头进行溯源，定位攻击报文的真实源头，主要采用报文标记、路由器日志等方式找到攻击报文源头，典型代表有 PI，PPM[9] 和 DPM[10] 等。

1.4 本文用到的验证方法

本文使用的两种方法均属于源地址路径过滤验证机制，基于域内源地址验证架构[11]，在自治域内，在报文传播路径上用不同的过滤规则对报文进行过滤，以达到源地址验证目的，两种方案如下。

方案 1 绑定 MAC 地址和 IP 地址。

如今比较流行的联网方式是把主机通过局域网组织在一起，然后通过交换机、路由器和 Internet 相连，这样就出现了如何识别用户、防止盗用的问题。由于 IP 地址只是逻辑上的标识，可以随意修改，不能用来标识用户；而 MAC 地址是物理上的标识，固化在网卡里，是很难被冒用的。针对 IP 地址时常被盗用和伪造的问题，将 IP 和 MAC 地址绑定起来便是解决的办法之一，事先获得主机的 IP 地址、MAC 地址和路由器接口信息，将无状态的 IP 地址和底层的 MAC 地址、交换机 Port 绑定起来，形成（MAC 地址，Port，IP 地址的）过滤表项，根据过滤表对伪造 IP 源地址报文进行过滤。其中 Port 为交换机端口，即使有人盗用了 IP 地址和 MAC 地址，他也不可能拥有真实的路由连接端口，从而在物理通道上隔离盗用者。

如图 1 所示，交换机事先获得主机 A 的源地址 IPsrc，MAC 地址以及主机与交换机相连的端口 Port1，在交换机中添加一个（MAC1，Port1，IPsrc）过滤表项，当真实源地址的报文到达交换机时，检查过滤规则，验证源地址为真实，通过报文并转发；当伪造源地址报文到达交换机时，与过滤表项规则不匹配，验证失败，丢弃不匹配的报文。

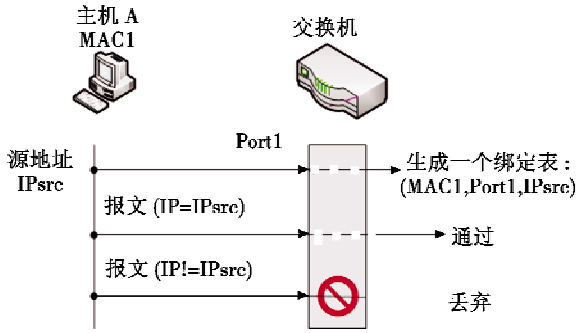

图 1 方案 1 原理图

Fig. 1 The schematic of scheme 1

将 MAC 地址和 IP 地址绑定，实现起来很简单，配置容易，不过当一台主机重新申请 IP 地址，或者设备更换网卡，造成 IP 地址和 MAC 地址变动时，需要向网络管理员提交更改后的地址，网络管理员做好登记，再手动进行更改绑定表，耗时且步骤繁琐。现有网络体系下，很难自动快速地更新（MAC，Port，IP）绑定表，这是该方案的不足之处。

方案 2 计算路径过滤。

在域内收敛的情况下，从某个源主机发往某目的主机的报文所走的路由路径是一定的，通过获取路由接口信息和邻居互联信息，进行路由计算，事先计算出域内某一台主机到另一台主机的路径，而伪造源地址的报文所走路径与事先计算的路径并不相符。如图 2 所示，从主机 A 发往主机 B 的数据包，计算出其所走的路由路径为：主机 A→交换机 1→交换机 2→交换机 3→主机 B，当主机 C 伪造主机 A 的 IP 地址向主机 B 发送数据包，所走实际路由路径为：主机 C→交换机 4→交换机 2→交换机 3→主机 B。对比可以发现，伪造源地址数据包所走路径和计算好的正确路径并不相符，因此在转发路径上可以用某种过滤规则过滤掉。

这是一个轻量级且效果显著的方案，过滤报文所增加的工作仅仅是在路由器中增加一条过滤规则和多进行一次或者两次路由表查找。如果仅仅是在单个路由器部署，过滤效果会很有限，因为一个路由器连接的链路有限，如果在域内多个路由器部署，对于抵抗 DDoS 攻击，效果将会十分明显，在自治域内 17% 交换机部署这个方案，可有效过滤 80% 的源 IP 地址伪造数据包。

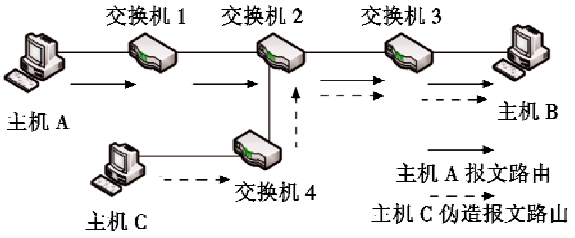

图 2　方案 2 原理图

Fig. 2　The schematic of scheme 2

在现有的网络体系架构下实现这个方案还面临着诸多困难。计算路径所需要的全局拓扑信息、链路状态信息、路由过滤表等，由于各个厂商的路由信息管理库规范不统一，从现有路由器上获取这些信息变得十分困难；计算路径后，配置路由表规则需要人工手动远程登录路由器，用命令行的方式配置路由表，效率低下且易出错；当域内主机或者交换机发生变化时，也缺乏一个及时有效的自动更新路由表的机制。因此，在现有的网络体系架构下，计算路径过滤方案实现起来难度很大。

2　SDN 特点及两个方案的阐述

2.1　SDN 的特点

SDN 将网络设备的控制面和转发面分离[12]，控制面独立出来成为单独的网络控制器，其上运行网络操作系统，向上为应用层提供可编程接口 API，从而构建开放的可编程网络环境；向下通过南向接口和虚拟化技术，统一管理网络设备[13]。图 3 所示的 SDN 架构图中，交换机利用基于安全连接的 OpenFlow 协议[14]与远程控制器通信。

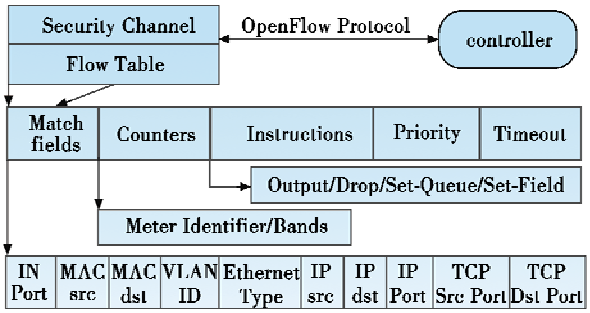

图 3　SDN 架构图

Fig. 3　SDN chart

基于 OpenFlow 的交换机需要维护流表这样一个重要的部件，其功能是进行数据包快速匹配和转发，相当于传统交换机的路由表。流表有包头、计数器、动作、优先级、超时等规则，包头进行相关的规则匹配，包括入端口、IP 地址、MAC 地址、以太网类型、VLAN ID 和 TCP 端口号等，计数器进行字节统计工作，动作则包括丢弃、转发、排队和设置域等[11]。控制器上运行网络操作系统，常见的控制器有 NOX，POX，Beacon，Floodlight 等（下文以 POX 为例进行讲解）。

SDN 这种全新的网络架构，如何实现上述两种源地址验证方案，能否克服两方案的不足，又能带来哪些优势呢？

2.2　方案 1：绑定 MAC 地址和 IP 地址

1）要进行 MAC 地址和 IP 地址绑定，首先要获取各个主机 MAC 地址、IP 地址和交换机接口 Port 信息。SDN 一大优势就是拥有全局视图能力，POX 控制器通过监听 Link_Event，Host_Event，Connection_Up，Connection_Down 事件，进行链路探测，发送 LLDP 报文，获取链路连接信息，保存各个主机和交换机相连的 Port 信息保存；通过 ARP 洪范机制进行 MAC 地址自学习，获取各个主机 MAC 地址；通过数据包第一次到达交换机的 Packet_in 消息，记录下各个主机的 IP 地址。

2）在 POX 控制器获取了以上信息后，用元组形式的数据结构在控制器中记录下每台主机的 MAC，IP，Port 信息，从而以 Flow_mod 的形式下发形式为（MAC，IP，Port）的三元组流表至交换机。

3）当同主机数据包再次到达交换机，进行流表查找和匹配工作，如果与（MAC，Port，IP）形式的流表不匹配，则丢弃报文，若匹配则进行转发。方案 1 流程如图 4 所示。

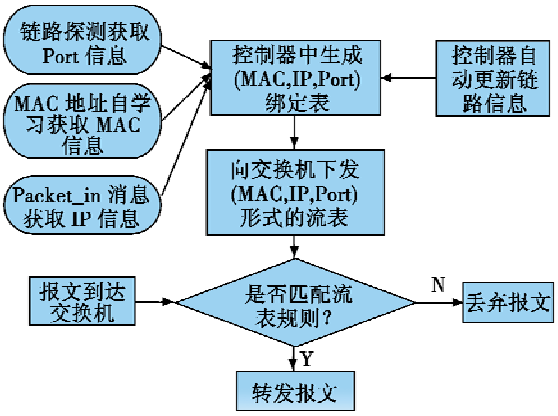

图 4　方案 1 流程图

Fig. 4　Flow chart of scheme 1

在 SDN 架构下，该方案实现简单，最大的优势是 POX 控制器可以监听链路状况，在有主机加入，IP 地址变更、MAC 地址变更和交换机变更等情况时，通过监听的 Link_Event，Host_Event，Connection_Down 等事件可以得到及时上报，并在控制器中自动更新（MAC，Port，IP）表。

2.3　方案 2：计算路径过滤

这个方案主要包含路由计算和流表生成两个部分。POX 控制器通过监听事件 Link_Event，Host_Event，Connection_Up，Connection_Down，获取域内整个拓扑路由器邻居互联信息，记录每个交换机、相邻交换机及相邻交

换机对应的连接端口;以最短路径算法为基础,计算出自治域内任意两个主机之间的路由路径,并给出该路径上经过每个交换机入口及对应的出口,形式为:host→(port,switch,port)→ … →host。其中,host 为主机,port 为交换机端口,switch 为交换机,向路径上的交换机下发形式为(IPsrc,IPdst,In_Port,Out_Port)的四元组流表,以此作为过滤规则,过滤伪造源地址报文。

路由计算采用最短路径 Floyd 算法,算法过程如下。

1)将拓扑中各交换机依次编号为 1,2,3,…,n,计算邻接矩阵 $\boldsymbol{D}_0$,其中,d_{ij}代表交换机 i 到交换机 j 的距离,如果两交换机之间没有链路相连通,则令 $d_{ij}^0=\infty$。

2)对于每一对交换机 i 和 j,看看是否存在一个节点 m 使得从 i 到 m 再到 j 比已知路径 d_{ij}^{m-1}更短,如果有则更新它,将最短路径记为 d_{ij}^m,相应递推式为

$$d_{ij}^m=\min\{d_{im}^{m-1}+d_{mj}^{m-1},d_{ij}^{m-1}\}。\tag{1}$$

每当计算一个节点,就记下它所走的路径,当算法结束时,矩阵 $\boldsymbol{D}_n$元素 d_{ij}表示交换机 i 到交换机 j 的最短距离。

Floyd 算法计算核心代码如下所示。

```
def Floyd:
    n = len(switchs)
    path = [([0] * range(n)) for i in range(n)]
    for i in range(n):
        for j in range(n):
            path[i][j] = i
for k in range(n):
    for i in range(n):
        for j in range(n):
            if d[i][j] > d[i][k] + d[k][j]:
                d[i][j] = d[i][k] + d[k][j]
                path[i][j] = path[k][j]
return d,path
```

代码中:“d[i][j]”代表交换机 i 到交换机 j 的距离;“switchs”为路由器顶点列表;“path”为路径矩阵。

其中,对于矩阵 $\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_n$对角线的元素都无需计算,并且,对所有的 $i=1,2,\cdots,n$,$d_{im}^{m-1}=d_{im}^m$,$d_{mi}^m=d_{mi}^{m-1}$,所以在矩阵 $\boldsymbol{D}_k$的计算中,第 k 行和第 k 列均无需计算,在矩阵 $\boldsymbol{D}_m$中,不仅是对角线元素无需计算,第 m 行和第 m 列也无需计算,所以总共仅需计算 $(m-1)\times(n-2)$个元素。

Floyd 算法必须计算 N 个矩阵 $\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_n$,其中每个矩阵包含 N^2 个元素,因此总共要计算 N^3 个元素,并且每一次计算都要做一次加法运算和一次取较小值运算,所以算法时间复杂度为 $O(2n^3)$,计算初始化路径矩阵“path[ ][ ]”时间复杂度为 $O(n^2)$,算法空间复杂度为 $O(n^2)$,Floyd 算法可以用迭代 N 次的 Dijkstra 算法来代替,但 Floyd 算法实现起来代码比 Dijkstra 算法精简一些。

3)流表生成和下发。根据源地址和目的地址计算出路径,向路径上的交换机下发(IPsrc,IPdst,In_Port,Out_Port)形式的流表。

```
def Install_path():
    p = get_path(self,IPsrc,IPdst)
    for switch in p:
        self. install(IPsrc,IPdst,in_port,out_port)
```

方案 2 不仅计算路由还生成流表,将两者结合起来,一步到位,生成的流表不仅可以实现路由转发,功能还可以验证过滤源地址。相比于方案 1,能进行路由选路算法,功能更加完备。

3 实验和结果分析

本文针对这两个源地址验证方案,进行仿真实验,实验环境如下:PC 为 i3 M330 @2.13 GHz CPU,内存 2 G;操作系统为 Ubuntu 12.04 LTS;Python 2.7,POX 0.2 carp,Mininet 2.1.0。

本文使用虚拟化网络仿真平台 Mininet,它可虚拟数百个节点的虚拟网络,节点类型和拓扑结构可以自定义,并且支持 OpenFlow 协议,具体网络拓扑如图 5 所示。

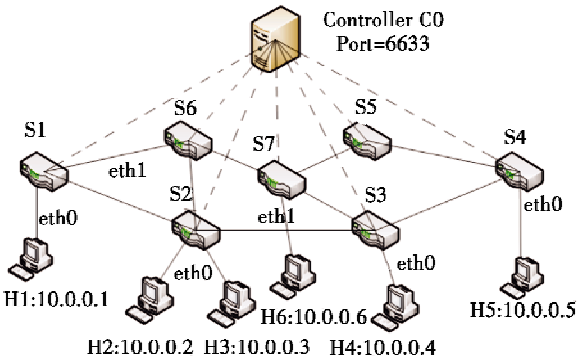

图 5 Mininet 实验拓扑

Fig. 5 Mininet test topology

在 POX 控制器上运行针对两个方案编写的 Python 代码,以发包脚本和 Mininet 自带发包机制为输入,进行 3 种测试:1)各主机之间正常发送 IP 包;2)伪造拓扑内主机源地址发送 IP 包;3)随机伪造源地址发送 IP 包。将测试的 10 次数据取平均值。

表 1 所示为两个方案在数据包初次到达和数据包正常转发的时延情况,从表 1 中可知,两方案在数据包初次到达比正常转发延时分别长 1.001 ms 和 3.215 ms,其原因为初始时交换机没有流表,需要将数据包以 Packet_in 的形式提交至控制器,控制器依据相应策略生成流表,之后数据包正常转发;数据包初次到达,方案 2 比方案 1 时延长 2.542 ms,由于方案 2 数据包初次到达交换机,被移交至控制器,控制器解析数据包包头,提取源目的地址,进行 Floyd 选路算法,而方案 1 控

制器中只需进行统计、生成流表，无需进行路由选路，所以时延稍小。

表 1　两方案时延对比

Table 1　Time delay comparison of the two schemes

ms

实验方案	平均时延	
	数据包正常转发	数据包初次到达
方案 1	0.024	1.025
方案 2	0.029	3.214

比较两种方案生成流表数目，如图 6 所示，从图中可以看出，方案 2 比方案 1 生成流表数目多，是因为方案 1 流表数目和主机数目相关，本实验拓扑主机数目较少，但当整个域内主机数目增加时，流表数目将会相应地增加；方案 2 流表数目在各交换机之间相差大，交换机 S2，S3 相比于交换机 S1，S5 更多，是由于选路算法导致，居中的交换机在多条路由路径上，并且还要生成反向流表，因此流表数目较多；在域内主机数目逐渐变大时，方案 1 流表数目将会超过方案 2。

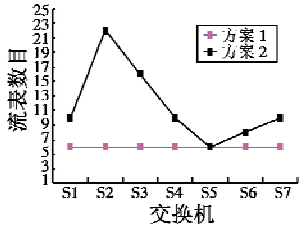

图 6　两方案流表数量对比

Fig. 6　Flow table quantities comparison of the two schemes

4　结束语

本文利用现有的两种源地址验证方法，在 SDN 架构下进行重新设计和实现，利用可编程控制器 POX 和虚拟化实验平台 Mininet 进行验证，根据实验结果对两种方案进行对比。方案 1，在域内主机数量增多时流表数目会明显增大，需要设计新的流表下发策略，在关键节点下发，减少流表规模；方案 2，在网络节点复杂时可能出现误判、漏判等，需要优化选路算法，这些将是后续工作的重点。

SDN 还处于发展初期，架构、协议、接口、应用等都处在研究阶段，本文将传统源地址验证技术与 SDN 相结合，在 SDN 架构上实现源地址验证，以期为 SDN 的发展做一点有益的探索。

参 考 文 献

[1] MICKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM Sigcomm Computer Communication Review, 2008, 38(2): 69-74.

[2] 姚广，毕军. 互联网中 IP 源地址伪造及防护技术[J]. 电信科学，2008，24(1)：26-33. (YAO G, BI J. Source address spoofing and prevention technologies in internet [J]. Telecommunications Science, 2008, 24(1): 26-33.)

[3] BREMLER B A, LEVY H. Spoofing prevention method [C]//The 24th Annual Joint Conference of the Computer and Communications Societies, Proceedings, IEEE, 2005: 536-547.

[4] FERGUSON P, SENIE D. ITEF RFC 2827 network ingress filtering: defeating denial of service attacks which employ IP source address spoofing[S]. Washington: ISOC, 2000.

[5] BOLTON C, LOWE G. Analyses of the reverse path forwarding routing algorithm[C]//The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Florence, 2004: 485-491.

[6] PARK K H, LEE H J. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets[C]//Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, New York: ACM, 2001: 15-26.

[7] CHENG J, WANG H N, KANG G S. Hop-count filtering: an effective denfense against spoofed DDoS traffic[C]//Proceedings of the 10th ACM Conference on Computer and Communications Security, New York: ACM, 2003: 30-41.

[8] WU J P, BI J, LI X. IRTF RFC5210 a Source Address Validation Architecture (SAVA) test bed and deployment experience[S]. Washington: ISOC, 2008.

[9] SAVAGE S, WETHERALL D, KARLIN A, et al. Network support for IP traceback[J]. IEEE/ACM Transactions on Networking, 2001, 9(3): 226-237.

[10] ABRAHAM Y, PERRIG A, SONG D. Pi: a path identification mechanism to defend against DDoS attacks[C]// Symposium on Security and Privacy, Proceedings, IEEE, 2003: 93-107.

[11] WU J P, REN G. Building a next generation Internet with source address validation architecture[J]. Science in China, 2008, 38(10): 1583-1593.

[12] CHUNG S, LIAO L, WAN J. Software defined networks [J]. Communications Magazine, IEEE, 2013, 51(2): 113-117.

[13] WEN F X, YONG G W, CHUAN H F, et al. A survey on software defined networking[J]. IEEE Communications Surveys & Tutorials, 2014, 17(1): 27-51.

[14] 左青云，陈鸣，赵广松，等. 基于 OpenFlow 的 SDN 技术[J]. 软件学报，2013，24(5)：1078-1097. (ZUO Q Y, CHEN M, ZHAO G S, et al. Research on OpenFlow based SDN technologies[J]. Journal of Software, 2013, 24(5): 1078-1097.)