第 46 卷　第 1 期　　计　算　机　工　程　　2020 年 1 月
Vol. 46　No. 1　　Computer Engineering　　January 2020

·网络空间安全·　　文章编号：1000-3428(2020)01-0121-08　　文献标志码：A　　中图分类号：TP393

基于 SDN 的 UDP 反射攻击响应方案

丁　伟[a]，张千风[a]，周文烽[b]

（东南大学 a. 网络空间安全学院；b. 计算机科学与工程学院，南京 211189）

摘　要：针对字符发生器协议、域名系统协议、网络时钟协议、简单网络管理协议、简单服务发现协议这 5 种类型的用户数据报协议（UDP）反射攻击放大器，提出基于入侵检测系统（IDS）的 UDP 反射攻击响应方案。在定位到反射攻击放大器的前提下，结合网络边界的软件定义网络技术，采用基于 OpenFlow 流表的响应规则对控制命令报文进行过滤，从而阻止 UDP 反射攻击。在中国教育和科研计算机网南京主节点的网络边界上的测试结果验证了该响应方案的可操作性和有效性。

关键词：用户数据报协议；反射攻击放大器；软件定义网络；反射攻击响应；网络边界

开放科学（资源服务）标志码（OSID）：

中文引用格式：丁伟，张千风，周文烽. 基于 SDN 的 UDP 反射攻击响应方案[J]. 计算机工程，2020，46(1)：121-128.

英文引用格式：DING Wei，ZHANG Qianfeng，ZHOU Wenfeng. UDP reflection attack response method based on SDN[J]. Computer Engineering，2020，46(1)：121-128.

UDP Reflection Attack Response Scheme Based on SDN

DING Wei[a]，ZHANG Qianfeng[a]，ZHOU Wenfeng[b]

（a. School of Cyber Science and Engineering；b. School of Computer Science and Engineering，Southeast University，Nanjing 211189，China）

【Abstract】 Based on an Intrusion Detection System（IDS），this paper proposes a response scheme for User Datagram Protocol（UDP） reflection attacks from 5 kinds of UDP reflection attack amplifiers，including Character Generator Protocol（CharGen），Domain Name System（DNS），Network Time Protocol（NTP），Simple Network Management Protocol（SNMP） and Simple Service Discovery Protocol（SSDP）. After the reflection attack amplifier is located，the scheme combines Software Defined Network（SDN） on the network boundary with response rules based on OpenFlow tables to filter control command messages，so UDP reflection attacks can be prevented. Test results on the network boundary of Nanjing main node of China Education and Research Computer Network（CERNET） demonstrate the operability and effectiveness of the proposed response scheme.

【Key words】 User Datagram Protocol（UDP）；reflection attack amplifier；Software Defined Network（SDN）；reflection attack response；network boundary

DOI：10.19678/j.issn.1000-3428.0053065

0　概述

用户数据报协议（User Datagram Protocol，UDP）是开放式系统互联（Open System Interconnection，OSI）参考模型中一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务。2013 年 3 月，世界反垃圾邮件组织 Spamhaus 遭受了峰值达到 300 Gb/s 的反射型 DDoS 攻击[1]，攻击者向互联网上开放的域名系统（Domain Name System，DNS）服务器发送了对 ripe.net 域名的 ANY 型解析请求，并将请求的源 IP 伪造成 Spamhaus 的 IP 地址。在此次攻击中，DNS 请求数据的长度为 36 Byte，而响应数据的平均长度为 3 000 Byte，攻击者将攻击流量放大了近 100 倍。

2018 年 1 月，NETSOUT A 发布了《13th Annual Worldwide Infrastructure Security Report》[2]，内容显示 DDoS 攻击对于互联网来说仍是主要的威胁。在 2017 年的所有攻击中，DDoS 占了 87%，其中大部分

基金项目：国家自然科学基金（61602114）；国家重点研发计划（2018YFB180202）。

作者简介：丁　伟（1962—），女，教授、博士，主研方向为网络安全、网络测量；张千风、周文烽，硕士研究生。

收稿日期：2018-11-07　　**修回日期：**2019-02-27　**E-mail：**wding@njnet.edu.cn

是 UDP 反射攻击[3]，并且 DNS 和网络时钟协议(Network Time Protocol，NTP)是被使用最多的协议。

UDP 反射攻击利用了 UDP 协议的无连接特性和服务协议自身的请求与响应之间数据量不对称的特性。攻击者通过伪造被攻击者地址请求这些使用 UDP 协议支持传输的服务，产生放大数倍的响应流量，从而达到攻击的目的[4]。那些提供 UDP 服务的主机通常被称为放大器，导致放大器发出反射攻击流量的报文通常被称为请求报文。本文针对 UDP 反射攻击放大器，提出基于入侵检测系统(Intrusion Detection System，IDS)的 UDP 反射攻击响应方案。

1 UDP 反射攻击的检测与响应

由于 UDP 反射攻击的流量特征明显且放大器使用真实地址，因此 UDP 反射攻击的检测在主干链路上比较容易实现，同时可以定位发出攻击流量的放大器。文献[5]给出一个以主干网边界流记录为数据源的检测算法，其核心思路是在网络边界路由器(检测点)使用端口和流量强度阈值约束并检出攻击流量。根据检测点提供的流记录，对基于 UDP 协议传输的数据源端口使用 UDP 反射攻击协议的端口匹配，对匹配到的流量再使用宿地址作为键对流量进行整合，强度大于阈值的整合流量被判定为最终的攻击流量。该算法实现简单，基于一个面向流记录的精细化网管平台(NBOS)[6]，其被部署到中国教育和科研计算机网(China Education and Research Network，CERNET)38 个主节点后运行效果良好，在 19、53、123、161 和 1900 5 个端口检测到了大量的反射攻击实例，同时定位了 CERNET 中5 个端口的放大器[5]。这 5 个端口对应的服务是字符发生器协议(Character Generator Protocol，CharGen)、DNS、NTP、简单网络管理协议(Simple Network Management Protocol，SNMP)、简单服务发现协议(Simple Serve Discovery Protocol，SSDP)，根据 NETSOUT A 的报告[2]，2017 年全球范围内这5 个端口的放大器占全部放大器的 94.1% 。

UDP 反射攻击的响应主要包括对放大器的防护、过滤伪造源地址数据包和清洗攻击流量 3 种方式[7]。放大器的防护是通过对放大器进行相应操作，使其无法成为 UDP 反射攻击的放大器，这些操作指的是关闭放大器上导致 UDP 反射攻击的服务或者对其版本进行升级进而使其不再支持该服务，如 NTP 中的 monlist 服务。这种响应方法需要有放大器的管理权限，而网络管理者和互联网服务提供商(Internet Service Provider，ISP)一般没有该权限。

由于 UDP 反射攻击中，攻击请求数据包与正常数据包是无法区分的，因此过滤伪造源地址数据包响应方法也不可行。清洗攻击流量利用 Anycast 技术[8]，将 DDoS 攻击流量传输到最近的清洗中心，使攻击流量得到有效稀释，从而对其进行阻断。该方法未针对反射攻击的特征，而是将其看成是普通的 DDoS 攻击进行流量清洗，效率较低。

文献[3]面向网络中定位到的放大器，提出一种基于访问控制列表(Access Control Lists，ACL)的拦截方式，通过在网络边界设备上设置 ACL 规则，拦截网内放大器特定漏洞服务协议端口上的相关报文，从而阻止 UDP 反射攻击。虽然该方法具有较好的效果，但其无法对正常使用协议的攻击报文进行拦截，例如 DNS 协议，由于该方法会导致这些服务失效，同时 ACL 的配置需要人工干预，因此不具备实际应用价值。

本文研究工作主要围绕反射攻击响应展开，从文献[3]的研究思路出发，提出反射攻击响应规则，在放大器已被定位的条件下，利用网络边界的软件定义网络(Software Defined Network，SDN)设备代替 ACL，采用响应规则对反射攻击请求报文进行过滤。在此基础上，实现对响应方案的整体流程设计，重点分析主干网在用型协议的反射攻击响应方案，并在 CERNET 南京主节点网络边界进行实现和测试。

2 基于 SDN 的反射攻击响应

2.1 SDN 技术与反射攻击响应

SDN 来源于斯坦福大学 Clean Slate 研究课题，本质特点是控制平面和数据平面的分离以及开放可编程性，优势在于可以对整个网络状态有所了解，为反射攻击请求报文的过滤提供了全局统一控制能力。

SDN 控制层中的控制器通过 OpenFlow 协议对 OpenFlow 交换机中的流表进行控制来达到对流量进行管理的目的[9]。OpenFlow[10]中的流表由一条条流表项组成，一条流表项由 7 个部分组成，分别是匹配字段(Match Fields)、优先级(Priority)、计数(Counters)、指令集(Instructions)、超时时间(Timeouts)、备注(Cookie)以及标识(Flags)。

SDN 技术中流表项包含本文使用的十元组：

$$[\text{src\_ip},\text{src\_mask},\text{src\_port},\text{dst\_ip},\text{dst\_mask},\text{dst\_port},\text{ip\_protocol},\text{submit\_time},\text{hard\_time},\text{action}] \tag{1}$$

其中，src_ip、src_mask、src_port、dst_ip、dst_mask、dst_port、ip_protocol、submit_time、hard_time、action 分别对应的语义是源地址、源地址掩码长度、源端口、目的地址、目的地址掩码长度、目的端口、协议号(17 表示 UDP，6 表示 TCP)、响应规则提交时间、响应规则执行持续时间和执行动作，如“转发”

和“丢弃”。

SDN 控制器可以根据接收到的响应规则,自动生成对应的 OpenFlow 流表项。流表项中的匹配字段由响应规则中前 7 项组成,决定了哪些报文将被该规则处理[11];指令集对应响应规则中的 action,决定匹配成功的报文将被如何处理;超时时间对应响应规则中的 submit_time 和 hard_time,决定了规则有效期;其他字段由 SDN 控制器自动生成。

SDN 对报文的控制能力超过普通路由器,所有 ACL 可实施的操作都可以在 OpenFlow 环境下完成,同时 OpenFlow 的转发操作提供了进一步的操作空间,本文在此基础上支持反射攻击响应。

2.2　研究环境与条件

本文研究在文献[3]的基础上展开,文献[3]通过在网络边界设备上设置 ACL 规则来拦截网内已定位放大器特定漏洞服务协议端口上的相关报文,从而阻止 UDP 反射攻击,但其无法对正常使用协议的攻击报文进行拦截,同时 ACL 的配置需要人工干预,因此不具备较好的普遍性。本文尝试利用网络边界的 SDN 设备代替 ACL 列表拦截发往放大器的攻击请求报文。假设研究条件为:1)放大器已被定位,是一个二元组(amp_ip,amp_port),其中,amp_ip 是放大器的 ip 地址,amp_port 是放大器提供反射协议服务的端口号;2)请求报文通过网络边界,这需要放大器与攻击者位于网络两侧,而与被攻击者的位置无关。在攻击者、放大器和被攻击者与网络边界构成的 8 种关系模型中(如图 1 所示),只有其中 4 种关系模型的攻击主机发往放大器的攻击请求报文通过网络边界,这 4 个场景分别为场景 a、场景 c、场景 e、场景 f,其中,场景 a 与场景 e 是对称的,场景 c 和场景 f 是对称的,下文仅面向场景 a 和场景 c 进行阐述。

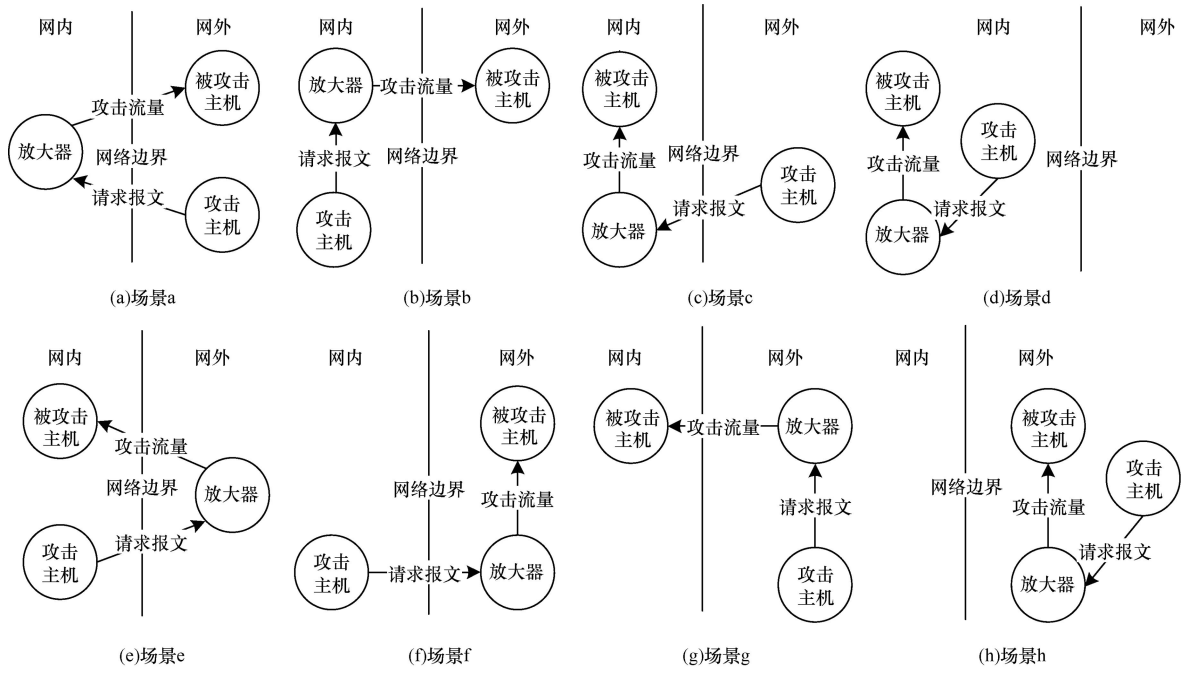

图 1　攻击者、放大器和被攻击者与网络边界的关系模型

Fig. 1　Relationship model between the attacker, amplifier, attacked host and network boundary

2.3　响应方案

文献[3]基于 ACL 规则能够较好地对 UDP 反射攻击进行拦截,其拦截规则也相对简单,这些规则可以与 OpenFlow 流表直接匹配,对应操作是“丢弃”。但该方案存在两方面的问题:1)这些拦截只能面向主干网不在用协议的攻击请求报文,对于正常使用的协议攻击请求报文不能做此类处理;2)ACL 规则的调整制定需要人工干预,如出现放大器集合变化此类情况时,需人工在网络边界重新设置规则。

本文利用 SDN 技术对反射攻击请求报文过滤提供的全局统一控制能力,基于 Openflow 流表设计一个自动化响应方案,其核心思想为:

1)对所有主干网不在用协议,采用文献[3]的 ACL 拦截规则和“丢弃”操作生成流表项进行响应,从流量中过滤反射攻击请求报文。流表项对应式(1)中的十元组为:

[any_ip,32,any_port,amp_ip,32,amp_port,17,
submit_time,0,“丢弃”]　　(2)

其中,src_ip 字段为 any_ip,表示任意 IP 地址,src_port 字段为 any_port,表示任意端口号,dst_ip 字段为 amp_ip,表示放大器 IP 地址,dst_port 字段为 amp_port,表示放大器提供的主干网不在用协议服务端

口,ip_protocol = 17 表示使用 UDP 协议,hard_time = 0 表示响应规则执行没有时间限制,该响应规则对应的流表项只能由用户主动撤销。

2)对所有主干网在用协议,将所有向放大器发出请求的报文,利用 OpenFlow 交换机的转发操作,将其发送给一个在后台运行的应用程序,这个程序对报文负载进行分析来确定是否为攻击请求报文。如果不是,则将其“回注”到流量中。对应的流表响应规则为:

[any_ip,32,any_port,amp_ip,32,amp_port,17,submit_time,0,“转发”] (3)

其中,dst_ip 字段为 amp_ip,表示放大器 IP 地址,dst_port 字段为 amp_port,表示放大器提供主干网在用协议服务的端口。当符合转发流表项的报文被转发到后台运行的应用程序后,后台应用程序会根据不同的反射协议对其应用不同的过滤规则进行“丢弃”或者“回注”操作。

3)根据放大器库的更新周期,实时动态调整面向流表的响应规则。

整个响应方案如图 2 所示,具体流程为:

1)响应规则生成与提交部分中的响应规则生成程序从放大器库中获取放大器信息,根据放大器库的更新,动态更新响应规则,并将可用或者待撤销的响应规则实时发送给基于 SDN 技术的流量管理系统。

2)基于 SDN 技术的流量管理系统中根据响应规则对通过网络边界的放大器请求报文进行转发或者丢弃处理,基于 SDN 技术可解决静态配置规则的问题。

3)报文过滤部分中的应用程序从报文过滤规则库中获取过滤规则,根据这些规则对接收到的转发报文进行“丢弃”或者“回注”操作。

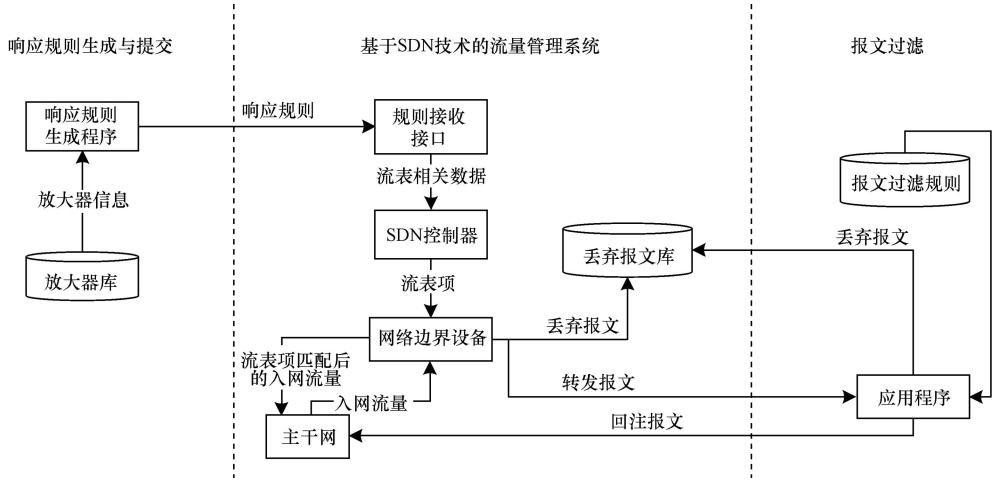

图 2 响应方案设计流程

Fig. 2 Design process of the response scheme

2.4 报文过滤规则

本节讨论应用程序中的报文过滤规则。这些过滤规则面向报文,目的在于响应主干网在用协议的反射攻击,处理对象是发往放大器且符合式(3)响应规则的报文,采用深度报文检测(Deep Packet Inspection,DPI)方式完成,导致反射攻击的服务协议彼此间没有关联,而且原因各不相同,因此过滤规则只能面向不同的协议分别进行讨论。由于 DNS 和 NTP 是目前最常见的在用反射协议[12],因此本节分别对相应的过滤规则进行讨论。这两个服务对应的端口分别为 53 和 123。

对于 NTP 而言,根据文献[7]导致反射攻击发生的原因是协议中的 monlist 请求。NTP 服务器响应 monlist 指令后会返回与其进行过时间同步的最近 600 个客户端的 IP 地址。响应包按照每 6 个 IP 进行分割,最多一个 NTP monlist 请求会形成 100 个响应包,实现较大的流量放大效果,因此基于 NTP 协议的 UDP 反射攻击会向放大器发送 monlist 请求报文。应用程序对转发过来的 NTP 报文,只要分析其是否为 monlist 请求报文即可,即匹配报文中是否含有“monlist = 1”,如果有,则该报文是 UDP 反射攻击请求报文,对其进行“丢弃”操作;否则对其进行“回注”操作。该过滤规则可以写为:

monlist = 1 (4)

对于 DNS 而言,根据相关研究,笔者认为导致反射攻击的报文应具有以下特征:

1)domain ∈ DNSSEC,表示域名参数 domain 是支持 DNSSEC 的域名。DNSSEC 在现有的 DNS 协议的基础上,对资源记录采用 RSA 加密算法,RSA 在加密算法中的密钥长度为 2 048 bit(256 Byte)。在 DNS 反射攻击中,攻击者对支持 DNSSEC 的域名进行 ANY 查询会额外返回上述 4 种资源记录信息,

从而扩大回复报文的字节大小,起到流量放大的作用。对于普通的非加密域名的解析一般为一个 IP 地址,不会产生大的流量。该特征来源于文献[13],该文献对 DNS 协议被用于 UDP 反射攻击的潜能性进行详细的实验分析和讨论,结果表明 DNSSEC 对 UDP 反射攻击是非常有利的,而对常规域名的解析放大效果不理想。

2)RR = ANY,表示查询类型是 ANY 类型,即向 DNS 服务器请求获取所有资源记录,ANY 查询会返回查询域名的所有类型资源记录信息,该类查询相比于其他查询类型流量放大效果更好。该特征来源于文献[14],该文献根据 US-CERT 观察到的 UDP 反射攻击案例,发现绝大多数攻击使用的都是 ANY 查询。另外,文献[15]的研究表明 DNS 协议中的 ANY 查询用于调试和测试,并且应用范围不广泛,目前已知的只有 qmail 应用和版本为 36.0 到 36.0.1 的 Firefox 应用会使用 ANY 查询,当 qmail 在对传出消息的封装进行域规范化时会使用 ANY 查询,但文献[16]指出现代邮件传输代理(Mail Transfer Agent,MTA)不再用 qmail 进行 ANY 查询。无论是 qmail 还是 Firefox 目前均已更新至最新版本,在新的版本中均不使用 RR = ANY 这个查询条件。因为没有正常的应用使用该条件,所以该特征可以作为过滤条件。

3)EDNS0 = 1,表示使用 EDNS0 协议,即扩展后的 DNS 协议。在 DNS 反射攻击中,攻击者使用 EDNS0 设置能够处理的最大 UDP 报文的大小。文献[17]表明 EDNS0 摆脱了使用 UDP 传输的 DNS 回复报文的大小在 512 Byte 以内的限制,这是放大效果最大化的反射攻击请求必须满足的条件。

4)RD = 1,表示递归请求解析域名,在递归查询模式下,DNS 服务器接收到客户机的请求,需使用一个准确的查询结果回复客户机。文献[18]表明如果 DNS 服务器本地没有存储查询的 DNS 信息,那么该服务器会递归询问上层 DNS 服务器,并最终将返回的查询结果提交至客户机。与特征 3 一样,递归查询也是 DNS 反射攻击中的 DNS 请求报文必须满足的条件。

根据以上分析内容可知,应用程序对 DNS 协议转发报文的过滤规则为:

$$\text{domain} \in \text{DNSSEC} \ \&\& \ \text{RR} = \text{ANY} \ \&\& \ \text{EDNS0} = 1 \ \&\& \ \text{RD} = 1 \tag{5}$$

3　响应方案实现与实验结果

3.1　响应方案实现

为验证响应方案的有效性,本文实现了上述方案,并在 CERNET 南京主节点网络边界上进行测试。在此次实施过程中,SDN 功能由 HYDRA 系统支持,放大器由 NBOS 系统提供,它们同样位于 CERNET 南京主节点网络边界。

3.1.1　HYDRA 和 NBOS 系统

NBOS[3]可检测到网内的反射攻击行为,并且定位到放大器位置。NBOS 提供的放大器信息包括放大器 IP 地址、反射端口、首次检出时间以及末次检出时间等,信息的更新周期是 5 min。

HYDRA 系统目前还是一个原型系统,只能在流入 CERNET 南京主节点网络的方向上使用 OpenFlow 流表对镜像流量进行控制。该系统部署在 CERNET 南京主节点网络边界上,使用的 SDN 设备是 H3C S6300 交换机,系统提供统一的响应符合式(1)规范的规则接口(新增规则和删除规则在不同位置),可以自动将来自应用系统的响应规则转换成流表项(每次一条)。目前流表支持的操作有“转发”和“丢弃”。转发后的报文以每 5 分钟 1 个 pcap 文件的形式存储在特定位置,应用程序可以在此获取这些转发的报文。对每条操作为“丢弃”的流表项,HYDRA 系统统计丢弃报文数量。由于只对镜像流量进行模拟操作,因此 HYDRA 系统目前没有流量回注功能。

由于 NBOS 系统能够提供 CharGen、DNS、NTP、SNMP、SSDP 这 5 种类型的 UDP 反射攻击协议的放大器信息,因此面向这 5 种协议进行测试[19-20]。其中的 NTP 和 DNS 属于主干网在用协议,采用式(4)和式(5)两条规则进行响应,另外 3 个属于主干网不在用协议,使用规则式(2)进行响应。因此,需要提交给 HYDRA 系统的响应规则分别为:

1)CharGen 反射攻击:

[0.0.0.0,32, -1,amp_ip,32,19,17,submit_time,0,“丢弃”]

2)SNMP 反射攻击:

[0.0.0.0,32, -1,amp_ip,32,161,17,submit_time,0,“丢弃”]

3)SSDP 反射攻击:

[0.0.0.0,32, -1,amp_ip,32,1900,17,submit_time,0,“丢弃”]

4)DNS 反射攻击:

[0.0.0.0,32, -1,amp_ip,32,53,17,submit_time,0,“转发”]

5)NTP 反射攻击:

[0.0.0.0,32, -1,amp_ip,32,123,17,submit_time,0,“转发”]

其中,0.0.0.0 表示任意 IP 地址,amp_ip 是反射器 IP 地址(NBOS 提供),submit_time 表示响应规则提交时间,该字段由 HYDRA 系统根据响应规则接收的时间自行设置。

3.1.2 基于 NBOS 与 HYDRA 平台的实现

图 2 响应方案的实现基于现有的 NBOS 和 HYDRA 平台,其中放大器库由 NBOS 提供,SDN 控制器与网络边界设备由 HYDRA 提供。在此条件下,实现的响应方案还需要完成响应规则生成程序和处理转发报文的应用程序。在此次实现中,前者位于 NBOS 放大器库所在硬件平台,以在内存中维护一张放大器响应规则表为核心数据结构工作,程序基于 NBOS 放大器库,每天对这张表进行一次维护,然后将需要修改的响应规则发送到 HYDRA 的规则接口。程序实现流程如图 3 所示。其中的失效规则指放大器响应规则表中放大器活跃时间与当前时间相差 N 天的表项对应的响应规则,N 的缺省值为 3。

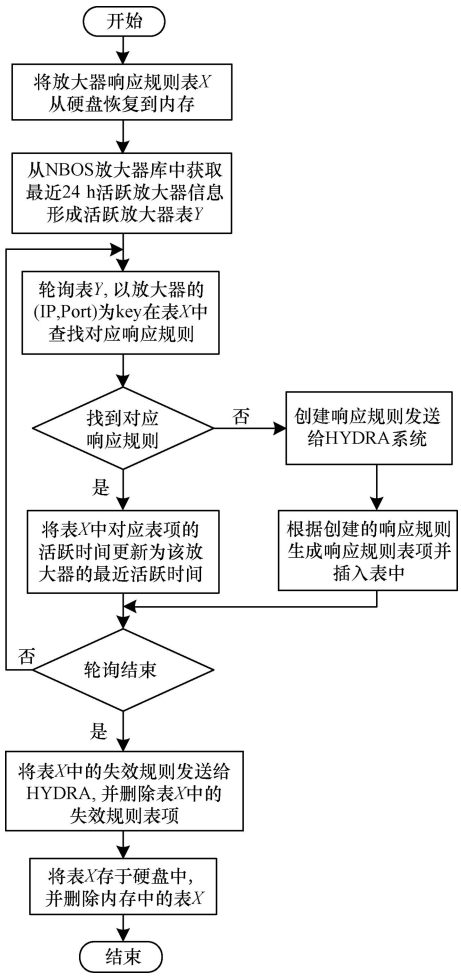

图 3 响应规则生成程序的实现流程

Fig. 3 Implementation flow of the response rule generator

应用程序目前是测试版,位于 HYDRA 系统中指定接收转发报文的主机上,只对两种主干网在用协议的转发报文进行分析,且已得出报文过滤规则,因此未考虑报文过滤规则的动态生成和更新,而是直接将规则式(4)和式(5)写在应用程序中。在此基础上,接收转发报文的主机将接收到的转发报文以每 5 分钟 1 个 pcap 文件的形式存储在硬盘上,应用程序从硬盘获取转发报文后,将符合规则式(4)和式(5)的报文放入丢弃报文库中。其余报文放入回注报文库中,由于 HYDRA 系统目前没有流量回注功能,因此应用程序未对回注报文作进一步处理。

3.2 实验结果

将响应规则生成程序和处理转发报文的应用程序在实际环境中进行测试,其中使用“丢弃”规则的 3 个主干网不在用协议的测试时间为 7×24 h,具体为 2018-04-24 05:00 至 2018-05-01 05:00。

由于 HYDRA 系统资源有限,因此对 DNS 与 NTP 两个主干网在用协议反射攻击的响应分别选择 5 台比较活跃的放大器进行 24 h 的响应实验,其中,DNS 反射攻击响应实施时间为 2018 年 4 月 26 日,NTP 反射攻击响应实施时间为 2018 年 4 月 27 日。

实验结果为响应规则生成程序针对 3 个主干网不在用协议共生成 25 条响应规则,其中,19 端口 2 条,1900 端口 5 条,其余 18 条面向 161 端口。针对 2 个主干网在用协议共生成 10 条响应规则,HYDRA 对这 35 条响应规则生成的 SDN 流表项的处理日志见图 4。其中,面向 123 端口和 53 端口的规则测试时间为 24 h,另外 15 条规则的测试时间是 7×24 h,packet_count 表示 HYDRA 系统拦截或者转发的报文数。

src_ip	dst_ip	src_port	dst_port	ip_protocol	submit_time	packet_count
0.0.0.0	210.28.[redacted].252	-1	19	17	2018-04-24 05:00:10	3235953
0.0.0.0	210.28.[redacted].12	-1	19	17	2018-04-24 05:00:10	3112681
0.0.0.0	210.28.[redacted].104	-1	53	17	2018-04-26 05:00:14	572275
0.0.0.0	210.29.[redacted].194	-1	53	17	2018-04-26 05:00:14	906050
0.0.0.0	58.193.[redacted].182	-1	53	17	2018-04-26 05:00:14	512963
0.0.0.0	210.28.[redacted].5	-1	53	17	2018-04-26 05:00:14	705703
0.0.0.0	210.29.[redacted].4	-1	53	17	2018-04-26 05:00:14	186746
0.0.0.0	202.112.[redacted].252	-1	123	17	2018-04-27 05:00:08	17987940
0.0.0.0	101.77.[redacted].204	-1	123	17	2018-04-27 05:00:08	19663683
0.0.0.0	101.77.[redacted].212	-1	123	17	2018-04-27 05:00:08	20407994
0.0.0.0	101.77.[redacted].190	-1	123	17	2018-04-27 05:00:08	18963775
0.0.0.0	101.77.[redacted].254	-1	123	17	2018-04-27 05:00:08	20919352
0.0.0.0	101.77.[redacted].54	-1	161	17	2018-04-24 05:00:10	159500
0.0.0.0	211.65.[redacted].32	-1	161	17	2018-04-24 05:00:10	132902
0.0.0.0	211.65.[redacted].170	-1	161	17	2018-04-24 05:00:10	141544
0.0.0.0	211.65.[redacted].209	-1	161	17	2018-04-24 05:00:10	142949
0.0.0.0	202.119.[redacted].66	-1	161	17	2018-04-24 05:00:10	16736
0.0.0.0	101.77.[redacted].29	-1	161	17	2018-04-24 05:00:10	348228
0.0.0.0	202.119.[redacted].82	-1	161	17	2018-04-24 05:00:10	25817
0.0.0.0	202.119.[redacted].11	-1	161	17	2018-04-24 05:00:10	24592
0.0.0.0	202.119.[redacted].13	-1	161	17	2018-04-24 05:00:10	24024
0.0.0.0	211.65.[redacted].172	-1	161	17	2018-04-24 05:00:10	133989
0.0.0.0	211.65.[redacted].173	-1	161	17	2018-04-24 05:00:10	207632
0.0.0.0	219.219.[redacted].140	-1	161	17	2018-04-24 05:00:10	245913
0.0.0.0	219.219.[redacted].139	-1	161	17	2018-04-24 05:00:10	16104
0.0.0.0	210.28.[redacted].56	-1	161	17	2018-04-24 05:00:10	224253
0.0.0.0	211.87.[redacted].100	-1	161	17	2018-04-24 05:00:10	578043
0.0.0.0	210.28.[redacted].12	-1	161	17	2018-04-24 05:00:10	578246
0.0.0.0	210.28.[redacted].104	-1	161	17	2018-04-24 05:00:10	21854
0.0.0.0	101.77.[redacted].2	-1	161	17	2018-04-24 05:00:10	424842
0.0.0.0	210.28.[redacted].99	-1	1900	17	2018-04-24 05:00:10	886701
0.0.0.0	219.219.[redacted].23	-1	1900	17	2018-04-25 05:00:09	2927458
0.0.0.0	219.219.[redacted].66	-1	1900	17	2018-04-24 05:00:10	5783977
0.0.0.0	202.119.[redacted].160	-1	1900	17	2018-04-24 05:00:10	640932
0.0.0.0	210.28.[redacted].133	-1	1900	17	2018-04-24 05:00:10	255395

图 4 UDP 反射攻击响应情况统计结果

Fig. 4 Statistics of responses to UDP reflection attacks

对于 CharGen、SNMP、SSDP 这 3 种主干网不在用协议,当其报文出现在主干网上时,一定是反射攻击请求报文,因此对实验结果中此类报文没有

误判的可能性。对于 DNS 和 NTP 这两种主干网在用协议,HYDRA 对其报文进行转发操作,供应用程序作进一步分析。应用程序对图 4 中转发报文的处理结果如表 1 和表 2 所示,其中第 1 列为放大器 IP 地址。

表 1　DNS 转发报文的处理结果
Table 1　Processing results of DNS forwarding messages

DNS 放大器	转发报文数	丢弃报文数	回注报文数	丢弃与转发报文比/%
210.28. ∗ .104	572 275	567 888	4 387	99.23
210.29. ∗ .194	906 050	809 110	96 940	89.30
58.193. ∗ .182	512 963	510 289	2 674	99.48
210.28. ∗ .5	705 703	584 887	120 816	82.88
210.29. ∗ .4	186 746	183 706	3 040	98.37
总计	2 883 737	2 655 880	227 857	92.10

表 2　NTP 转发报文的处理结果
Table 2　Processing results of NTP forwarding messages

NTP 放大器	转发报文数	丢弃报文数	回注报文数	丢弃与转发报文比/%
101.77. ∗ .254	20 919 352	20 919 350	2	99.99
101.77. ∗ .190	18 693 775	18 693 774	1	99.99
101.77. ∗ .204	19 663 683	19 663 682	1	99.99
101.77. ∗ .212	20 407 994	20 407 993	1	99.99
202.112. ∗ .252	17 987 940	17 987 939	1	99.99
总计	97 672 744	97 672 738	6	99.99

由表 1 和表 2 可知,DNS 和 NTP 转发报文的 90% 以上均是丢弃报文,即反射攻击请求报文,其余为回注报文,DNS 协议和 NTP 协议的丢弃报文分别符合规则式(4)和规则式(5),根据上文分析结果,符合这两条规则的报文若出现在主干网上,则为反射攻击请求报文,因此不存在误判的可能性。

由于 DNS 的回注报文数偏多,远大于 NTP 的回注报文数,且其报文过滤规则更复杂,因此本文任意选取一条 DNS 回注报文记录来查看其报文特征,判断是否为正常通信报文,具体情况如图 5 所示。

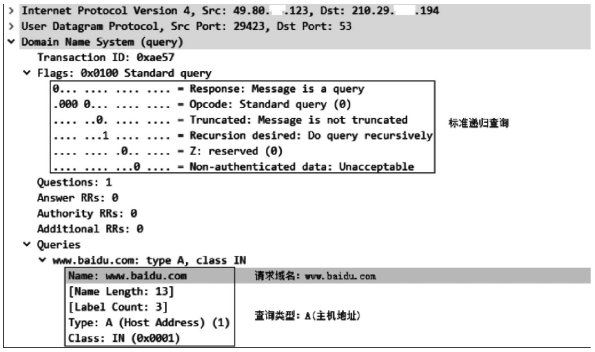

图 5　DNS 回注报文记录
Fig. 5　Records of reinjected DNS messages

由图 5 可知,该 DNS 回注报文是由 IP 为 49.80. ∗ .123的主机发往 IP 为 210.29. ∗ .194 的 DNS 放大器请求报文,请求查询域名 www.baidu.com 对应的 IP 地址记录,属于 DNS 正常通信报文,可以回注至主干网中。本文未进行性能方面的测试,这主要是因为 Hydra 平台除了 UDP 反射攻击响应系统外,还同时支持其他系统的工作。从本文实现系统的角度出发,由于响应原理是面向请求报文而不是攻击流量,因此本身就具有性能方面的优势,算法时间复杂度为 $O(m)$,其中 m 是发往网内请求报文的数量。

4　结束语

本文提出一种 UDP 反射攻击自动响应方案。该方案基于 SDN 设备在网络边界使用流表,对发往已定位放大器的控制命令进行拦截,在 CERNET 南京主节点网络进行 5 种反射攻击协议的测试验证了该方案的有效性和可操作性。目前互联网中大多数反射攻击均由 CharGen、DNS、NTP、SNMP、SSDP 这 5 种类型的 UDP 反射攻击协议产生,因此该方案如果在实际网络环境中进行应用,则可以有效控制 UDP 反射攻击。下一步将对更多的在用 UDP 反射攻击协议进行研究,如 C-LDAP、Memcached 等,探究基于软件定义网络的响应方案。

参考文献

[1] PRINCE M. The DDoS that knocked spamhaus offline (and how we mitigated it) [EB/OL]. [2018-10-15]. https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/.

[2] NETSCOUT's 14th annual worldwide infrastructure security report[EB/OL]. [2018-10-15]. https://www.netscout.com/report/.

[3] DING Wei, WANG Li, SU Qi. UDP reflection attacks interception based on ACL[J]. Journal of Huazhong University of Science and Technology (Nature Science Edition), 2016, 44(11): 21-25. (in Chinese)
丁伟,王力,苏琪. 基于 ACL 的 UDP 反射攻击拦截[J]. 华中科技大学学报(自然科学版), 2016, 44(11): 21-25.

[4] THOMAS D R, CLAYTON R, BERESFORD A R. 1000 days of UDP amplification DDoS attacks [C]// Proceedings of 2017 APWG Symposium on Electronic Crime Research. Washington D. C., USA: IEEE Press, 2017: 79-84.

[5] LI Gang. Research of scanning and DRDoS attack detection based on netflow [D]. Nanjing: Southeast University, 2016. (in Chinese)
李刚. 基于流记录的扫描和反射攻击行为主机检测[D]. 南京:东南大学, 2016.

[6] ZHANG Weiwei, GONG Jian, DING Wei. NBOS: a fine network management system based on flow technology[C]// Proceedings of the 19th Annual Conference of CERNET. Jinan: Management Committee of CERNET, 2012: 41-46. (in Chinese)

张维维,龚俭,丁伟,等. NBOS:一个基于流技术的精细化网管系统[C]//中国教育和科研计算机网 CERNET 第十九届学术年会论文集. 济南:中国教育和科研计算机网 CERNET 管理委员会,2012:41-46.

[7] TAO Naiyong. The principle and protection methods of DDoS amplification attack [J]. Telecommunication Network Technology,2017(10):89-93. (in Chinese)
陶乃勇. DDoS 放大攻击原理及防护方法[J]. 电信网技术,2017(10):89-93.

[8] JOHNSON D, DEERING S. Reserved IPv6 subnet anycast addresses:RFC2526[S]. IETF,1999:1-3.

[9] JARRAYA Y,MADI T,DEBBABI M. A survey and a layered taxonomy of software-defined networking[J]. Communications Surveys and Tutorials, 2014, 4(4): 1955-1980.

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

[11] LI Hefei, HUANG Xinli, ZHENG Zhengqi. Detection method of DDoS attack based on software defined network and its application[J]. Computer Engineering, 2016,42(2):118-123. (in Chinese)
李鹤飞,黄新力,郑正奇. 基于软件定义网络的 DDoS 攻击检测方法及其应用[J]. 计算机工程,2016,42(2):118-123.

[12] CHEN C C, CHEN Y R, LU W C, et al. Detecting amplification attacks with software defined networking[C]// Proceedings of IEEE Conference on Dependable and Secure Computing. Washington D.C.,USA:IEEE Press,2017:7-10.

[13] RIJSWIJK-DEIJ R V, SPEROTTO A, PRAS A. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study[C]//Proceedings of Conference on Internet Measurement. New York,USA:ACM Press,2014: 449-460.

[14] US-CERT. DNS amplification attacks[EB/OL]. [2018-10-15]. https://www.us-cert.gov/ncas/alerts/TA13-088A.

[15] MAREK M, OLAFUR G. Deprecating the DNS ANY meta-query type [EB/OL]. [2018-10-15]. https://blog.cloudflare.com/deprecating-dns-any-meta-query-type/.

[16] FINCH T. The qmail ANY query bugs [EB/OL]. [2018-10-15]. https://fanf.livejournal.com/122220.html.

[17] ANDREWS M. Extension Mechanisms for DNS (EDNS) EXPIRE option: RFC 7314 [S]. Internet Systems Consortium,2014:1-4.

[18] DAGON D, ANTONAKAKIS M, VIXIE P, et al. Increased DNS forgery resistance through 0x20-bit encoding:security via leet queries[C]//Proceedings of ACM Conference on Computer and Communications Security. Alexandria,USA:[s.n.],2008:211-222.

[19] BERTI-EQUILLE L, ZHAUNIAROVICH Y. Profiling DRDoS attacks with data analytics pipeline [C]// Proceedings of 2017 ACM Conference on Information and Knowledge Management. New York, USA: ACM Press,2017:1983-1986.

[20] PAXSON V. An analysis of using reflectors for distributed denial-of-service attacks[J]. ACM SIGCOMM Computer Communication Review,2001,31(3):38-47.

编辑 陆燕菲

(上接第 120 页)

[13] CHEN Hu, WEI Shimin, ZHU Changjie, et al. Secure certificateless aggregate signature scheme[J]. Journal of Software,2015,26(5):1173-1180. (in Chinese)
陈虎,魏仕民,朱昌杰,等. 安全的无证书聚合签名方案[J]. 软件学报,2015,26(5):1173-1180.

[14] DU Hongzhen,HUANG Meijuan,WEN Qiaoyan. Efficient and provably-secure certificateless aggregate signature scheme[J]. Acta Electronica Sinica,2013,41(1):72-76. (in Chinese)
杜红珍,黄梅娟,温巧燕. 高效的可证明安全的无证书聚合签名方案[J]. 电子学报,2013,41(1):72-76.

[15] LU Haijun,XIE Qi. An efficient certificateless aggregate signcryption scheme from pairings[C]//Proceedings of International Conference on Electronics,Communications and Control. Ningbo,China:[s.n.],2011:137-146.

[16] TU Hang, HE Debiao, HUANG Baojun. Reattack of a certif-icateless aggregate signature scheme with constant pairing computations [J]. Scientific World Journal, 2014(3):9-10.

[17] ZHANG Lei,ZHANG Futai. Security model for certificate-less aggregate signature schemes [C]//Proceedings of International Conference on Computational Intelligence and Security. Suzhou,China:[s.n.],2008:243-253.

[18] ZHANG Lei, QIN Bo, WU Qianhong, et al. Efficient many-to-one authentication with certificateless aggregate signatures [J]. Computer Networks the International Journal of Computer & Telecommunications Networking, 2010,54(14):2482-2491.

[19] YU Huifang, YANG Bo. Provably secure certificateless hybrid signcryption [J]. Journal of Software, 2015, 38(4):804-813. (in Chinese)
愈惠芳,杨波. 可证安全的无证书混合签密方案[J]. 计算机学报,2015,38(4):804-813.

[20] CHEN Yuchi,HORNG G,LIU Chaoliang,et al. Efficient certificateless aggregate signature scheme[J]. Journal of Electronic Science and Technology,2012,10(3):209-214.

[21] ZHONG Hong,HUANG Bo,CUI Jie,et al. Conditional privacy-preserving authentication using registration list in vehicular Ad hoc networks[J]. IEEE Access,2018(6): 2241-2250.

编辑 索书志