计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(9):143−150 [doi: 10.15888/j.cnki.csa.006559]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

面向软件定义网络的隐蔽通信检测机制①

倪永峰, 闫连山, 崔允贺, 李赛飞

(西南交通大学 信息科学与技术学院, 成都 611756)

摘　要: 为提高软件定义网络抵抗高级持续性威胁的能力, 对软件定义网络特性及高级持续性威胁中的隐蔽通信进行了分析, 提出了一种适用于软件定义网络的高效隐蔽通信检测机制. 该隐蔽通信检测机制首先利用软件定义网络抓取网络流量并从中获取可能包含隐蔽通信的报文; 随后从上述报文中提取 SSL 证书, 并计算用于表征该证书的特征值; 最后采用孤立森林算法对证书的特征值进行检测以判断证书是否为非法证书, 基于此检测结果判断网络中是否存在隐蔽通信. 实验结果及分析表明, 该隐蔽通信检测机制能够提高隐蔽通信检测精度, 降低隐蔽通信误检率; 同时该机制可扩展性较高, 能够适用于不同应用场景.

关键词: 软件定义网络; 隐蔽通信; 孤立森林; SSL 证书

引用格式: 倪永峰,闫连山,崔允贺,李赛飞.面向软件定义网络的隐蔽通信检测机制.计算机系统应用,2018,27(9):143–150. http://www.c-s-a.org.cn/1003-3254/6559.html

Covert Communication Detection Mechanism for Software Defined Network

NI Yong-Feng, YAN Lian-Shan, CUI Yun-He, LI Sai-Fei

(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: In order to detect advanced persistent threat in software defined network, an efficient mechanism utilized in SDN is proposed to detect covert communication in this study, based on analyzing the architecture of SDN and covert communication in advanced persistent threat. When detecting covert communication, this mechanism firstly captures the transmitted traffic from the underlying network. Subsequently, it extracts SSL certificates from the captured packets and calculates several eigenvalues of the extracted SSL certificates. At last, using isolation forest algorithm, it detects whether these SSL certificates are abnormal taking advantages of the extracted eigenvalues. Based on the detection result of SSL certificates, this mechanism can judge whether there is covert communication in this network. Experimental results verify that the proposed mechanism can improve the detection accuracy and reduce false positive of covert communication. At the same time, this mechanism has high scalability, which makes it easily implemented in other scenarios.

Key words: software defined network; covert communication; isolation forest; SSL certificate

作为一种新型网络架构, 软件定义网络 (Software Defined Network, SDN) 将网络中的控制功能与数据转发功能相分离, 实现了网络可编程化[1]. 在该网络架构中, 控制功能逻辑性地集中在控制器中, 其负责计算路由和下发流表. 数据转发功能由交换机实现, 其依据流表对数据报文进行转发. 当前 SDN 架构中, 控制器与交换机之间多采用 OpenFlow 协议进行通信. OpenFlow 协议是一种新型网络协议, 其定义了控制器与 SDN 交换机之间的通信方式及交换机处理到达报文的规则. 基于此协议实现的 SDN 网络能够对其内的网络设备进行集中管理[2]. 自 2009 年提出第一版本至今, OpenFlow 协议已有 5 个版本, 其中 1.3 版本的 OpenFlow 协议是

① 基金项目: 国家铁总重大项目 (2016X008-D)
Foundation item: Major Project of China Railway (2016X008-D)
收稿时间: 2018-01-19; 修改时间: 2018-03-13; 采用时间: 2018-03-14; csa 在线出版时间: 2018-08-16

当前使用最广泛的南向接口协议之一. SDN 架构使得控制器具备全局可见性, 因此其能够精细化地对流量进行调度. 基于此优势, 目前 SDN 已成功应用于众多数据中心[3,4].

高级持续性威胁 (Advanced Persistent Threat, APT) 是一种新型网络攻击, 其比传统攻击手段更具威胁性和破坏性[5]. APT 攻击持续时间通常可达数月甚至数年, 所使用的技术较传统攻击手段更为复杂, 同时较高的时间和金钱消耗决定了 APT 攻击通常用于企业、军工等重要目标. 因此美国将 APT 攻击纳为网络战的范畴, 并成立了专门的网络部队以应对和发起 APT 攻击. 自 2010 年 Google 遭受“极光”攻击以来, 高级持续性威胁便引起网络安全界的广泛关注. 经过长期研究, 众多研究者通常将 APT 攻击分为攻击准备、横向攻击、资料回传、退出四个阶段. 2011 年 Tankard 等人总结出了 APT 攻击的流程, 其中详细叙述了 APT 攻击中常用的水坑攻击以及端口扫描等攻击手段, 并建议采用日志分析、文件完整度检查、注册监控以及恶意病毒检测等技术检测 APT 攻击[6]. Li 等人总结了各攻击阶段常用的攻击手段: APT 进入阶段通常使用水坑攻击实现, 潜伏阶段多使用远程控制、横向移动、获取特权、隐蔽通信等手段, 退出阶段通常包括资料回传、销毁数据等步骤[7]. Chandra 等人研究了 APT 攻击开始阶段攻击者常用的社会工程学理论, 针对网络资源虚拟化框架 OpenStack 制定了纵深防御机制[8]. 由于 APT 攻击具有高隐蔽性, 传统的监测设备不易察觉其行为目的, Stefan Rass 等人提出了一种基于博弈论的 APT 检测机制, 该机制通过分析网络行为评估网络的风险性以检测 APT 攻击[9]. 针对 APT 攻击的特点, Ibranhim Ghafir 和 Vaclav Prenosil 提出了八种检测方法, 分别为恶意附件检测、恶意域名检测、C&C 隐蔽通信检测、恶意证书检测、鱼叉攻击检测、恶意文件哈希值检测、域名流量检测以及匿名网络 TOR 检测[10].

隐蔽通信是指受控主机与攻击者控制的 C&C 服务器 (Command and Control server) 建立的能够躲避网络监控的秘密通信, 是 APT 攻击各个阶段必须使用的技术. 无论是起初的横向攻击或获取资料后的回传过程, 为避免被网络中安全设备检测出异常, 攻击者都必须在受控主机与 C&C 服务器之间建立隐蔽通信信道. 文献[11]中研究发现常见 APT 攻击的隐蔽通信均采用 SSL/TLS 协议, 此类协议是加密协议并且采用证书的方式交换客户端与服务器使用的密钥. 因此如果能够检测网络中存在的恶意隐蔽通信就可以及时阻止攻击, 避免 APT 攻击造成的损失. Fu 等人提出了一种区分 SSL 流量的机制, 该机制采用报文长度、报文到达间隔以及流量方向作为特征值, 利用 C4.5 机器学习算法对 SSL 流量进行分类[12]. Ibranhim Ghafir 等人利用开源架构 Intelligence Framework[13]提出了一种 SSL 证书检测机制, 首先从互联网中获取 IF 架构证书黑名单并建立本地黑名单列表, 当有 SSL 流量时提取其中的服务器证书信息, 然后判断此服务器证书是否在本地黑名单中, 若不在本地黑名单中 SSL 报文才可通过[14]. 该方法依赖 IF 架构中的黑名单, 同时需要及时更新黑名单信息. Cao 等人提出了一种两步检测方法, 该方法首先检测 SSL/TLS 服务器可信度, 然后提取证书信息对证书包含的信息进行评价, 最后得到证书的安全等级[11].

为增强 SDN 抵抗 APT 攻击的能力, 本文对 APT 攻击的关键步骤-隐蔽通信-进行了分析及研究, 基于此, 本文提出了一种高效的隐蔽通信检测机制. 该机制利用 SDN 的特性便捷地抓取 SSL/TLS 流量, 并从中提取证书信息, 然后计算能够表征证书特征的特征值, 并将上述特征值输入证书可信度检测模块以判断是否存在隐蔽通信. 实验分析及结果表明本文的隐蔽通信检测方案可以准确区分正常证书和恶意证书, 进而检测出 SDN 网络中的隐蔽通信.

1 面向 SDN 网络的隐蔽通信检测机制背景

1.1 SDN 架构

当前 SDN 通常采用 OpenFlow 协议作为南向接口协议. OpenFlow 协议规定, 当交换机收到新到达的报文时, 其按照流表优先级依次匹配其内流表中的 MATCH 域. 如图 1 所示, MATCH 域包含网络层协议、运输层协议以及运输层协议端口等匹配项. 交换机一旦找到能匹配数据报文的流表, 就根据流表 ACTION 域中的端口将报文转发出去. 流表 ACTION 域中的端口可以为报文下一跳交换机与当前交换机所连接的端口, 也可为控制器与当前交换机连接的端口, 即交换机能够将报文转发至下一跳交换机或服务器, 也能够将报文转发至控制器. 若交换机不能找到匹配报文的流表, 则将此报文上报至控制器, 由控制器计算

报文在网络中的转发路径, 并下发此报文能匹配的流表至交换机.

在 SDN 架构中, 控制器具有全网拓扑可见性, 因此其能够精确控制流量转发路径: 将大流量从带宽较大的链路转发, 小流量从带宽较小的链路转发, 当流量增大时能够修改流表将数据从带宽较小的链路转至带宽较大的链路. 采用 SDN 架构除了可以对流量精确控制之外, 还能够通过部署在交换机上的流表抓取特定类型的报文, 而在传统网络中获取特定协议的流量通常需要进行流量的分类识别, 因此在有选择地抓取流量方面 SDN 架构具有较大的优势. 本文正是利用 SDN 的这一优势抓取所需要的报文.

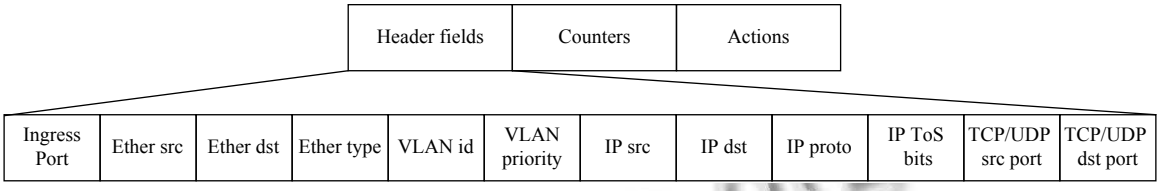

图 1 OpenFlow1.0 流表项结构

1.2 SSL 隐蔽通信流程

为增强网络抵抗 APT 攻击的能力, 本文分析了几种典型的 APT 攻击案例, 研究发现 APT 攻击中隐蔽通信检测的关键在于非法证书检测.

2009 年谷歌遭遇的极光行动是最典型的 APT 攻击, 在此次攻击中攻击者使用了 SSL 加密通信对受控服务器进行控制. 持续时间长达五年的暗鼠行动中, 攻击者使用了远程命令与控制通信远程控制受控服务器窃取攻击机密资料. 在 APT 攻击火焰病毒与高斯病毒中, 攻击者均采用自签名证书建立隐蔽. 由上述可知, 典型的 APT 攻击中攻击者均使用了隐蔽通信, 并且为躲避网络安全设备的监测通常采用 SSL 协议加密. 根据 SSL 协议, 通信双方传输数据之前需协商加密算法以及交换密钥, 这些信息均在数字证书 (Server Certificate) 中. 数字证书包含攻击者控制的 C&C 服务器的身份信息, 以及 C&C 服务器密钥和其支持的加密算法类型. 攻击者为防止自身信息暴露均会采用非法证书. 因此若能在隐蔽通信建立阶段检测出攻击者使用的非法证书, 就能检测出网络存在的隐蔽通信, 从而防止网络遭受进一步的攻击.

为获取隐蔽通信使用的非法证书, 需要对建立 SSL 通信的流程进行研究. SSL 通信由受控服务器中的恶意软件发起, 恶意软件根据其内部机制发起 DNS 请求, 获取 C&C 服务器 IP 地址, 进而建立 TCP 连接, 进入 SSL 握手阶段. 受控服务器发送 ClientHello 报文, C&C 服务器接收 ClientHello 报文之后, 回复带有数字证书的 ServerHello 消息, 其中数字证书包含 C&C 服务器的身份信息以及其使用的密钥, 然后受控服务器回复自身的密钥发送给 C&C 服务器. 最后, 受控服务器与 C&C 服务器采用协商一致的加密算法建立通信信道, 握手阶段结束. SSL 握手阶段结束之后, 受控服务器与 C&C 服务器进入正常通信状态.

综上所述, 采用 SSL 加密的隐蔽通信在建立通信的握手阶段均会使用非法证书, 同时由于合法加密通信使用的证书均是由可信机构颁发的合法证书, 因此网络中的合法通信不存在非法证书. 因此若能检测出网络中的非法证书, 就能检测出网络中存在的隐蔽通信. 本文提出的面向 SDN 网络的隐蔽通信检测机制正是检测网络中 SSL 协议使用的证书是否非法.

2 面向 SDN 网络的隐蔽通信检测机制

2.1 面向 SDN 的隐蔽通信检测机制 (SD-CCD)

如图 2 所示, 本文假设有 M 个交换机和 N 个服务器, 其中 CH 为受控服务器, AT 为 C&C 服务器 (Command and Control server), DS 为隐蔽通信检测服务器, Controller 为 SDN 控制器. 根据 SDN 架构, 交换机 $S=\{S_1, S_2, S_3, \cdots\}$均与控制器 Controller 连接.

受控服务器 CH 为攻击者通过社会工程学获取到权限的目标网络内的服务器, 其与 C&C 服务器建立连接并按照指令对目标网络发起扫描或嗅探以窃取网络中的资料. C&C 服务器 AT 为攻击者控制的服务器, 攻击者通过 C&C 服务器与受控服务器 CH 进行隐蔽通信, 控制 CH 在目标网络中的行动, 并且接收 CH 窃取的资料.

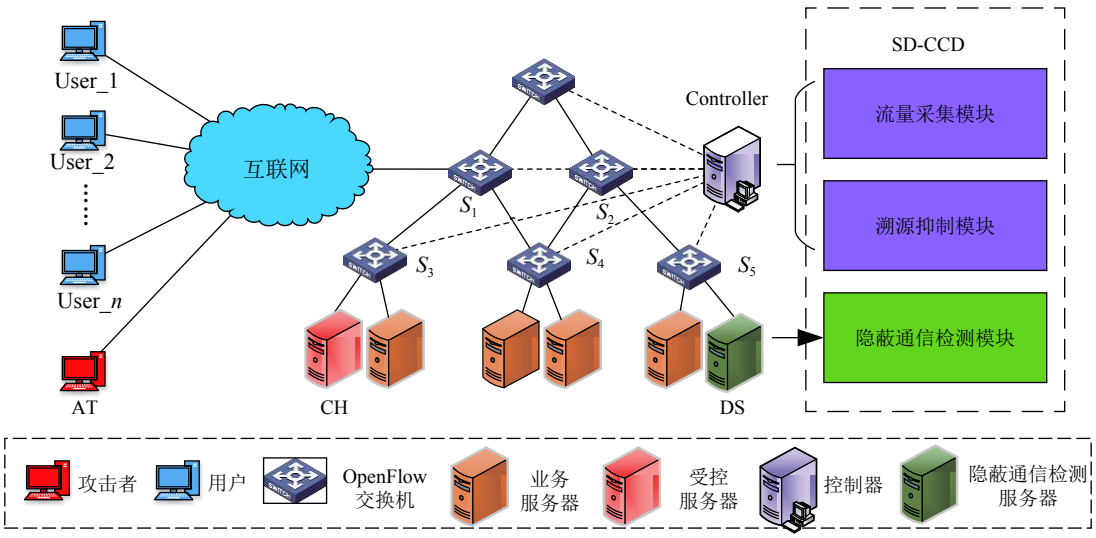

图 2 SDN 隐蔽通信检测架构

如图 2 所示, 本文提出的面向 SDN 的隐蔽通信检测机制 (Software-Defined Convert Communication Detection mechanism, SD-CCD) 由运行在隐蔽通信检测服务器 DS 上的隐蔽通信检测模块及运行在控制器上的流量采集模块及溯源抑制模块组成. 其中, 隐蔽通信检测服务器 DS 上运行隐蔽通信检测模块, 该模块包括证书提取、特征值计算以及隐蔽通信检测算法. 其主要对输入的 SSL 流量进行处理, 判断 SSL 流量是否为非法隐蔽通信. 若为非法隐蔽通信, 隐蔽通信检测服务器发送警报事件到控制器 Controller, 然后由控制器进行处理.

运行在控制器上的流量采集模块和溯源抑制模块负责 SSL 流量的采集及对非法隐蔽通信的溯源和抑制. 流量采集模块利用 OpenFlow 中的流表将 SSL 流量导入隐蔽通信检测模块 DS, 溯源抑制模块对受控服务器发起溯源并抑制其在网络中的行为. 控制器 Controller 中流量采集模块基于 SDN 特性利用 OpenFlow 流表对网络中的 SSL 报文进行采集. 虽然现行 OpenFlow 协议中还没有针对 SSL 协议的匹配项, 但是 SSL 协议封装于 TCP 协议中且通常使用 443 端口, 因此本文将使用 TCP 协议并且目的端口与源端口均为 443 的报文视为 SSL 报文. 当第一个 SSL 报文由交换机上报至控制器, 控制器计算完路由路径并下发流表时, 在流表 MATCH 域中添加运输层协议为 TCP 且源端口和目的端口均为 443 的匹配项, 同时在此流表的 ACTION 域中添加转发至隐蔽通信检测服务器所在的端口, 以此就可抓取所需的 SSL 报文.

在进行 APT 攻击时, 攻击者获取 CH 权限后, 需与 AT 建立 SSL 通信, 但此时 CH 所挂载的交换机 S_3 中没有转发 SSL 报文的流表, 因此不能转发 CH 发出的 ClientHello 报文. 根据 OpenFlow 协议, S_3 将 ClientHello 报文转发至控制器 Controller, 控制器计算出 CH 到 AT 的路由路径并依次下发流表至路径上的交换机, 同时计算出受控服务器 CH 到隐蔽通信检测服务器 DS 的路径并下发流表至路径上的交换机, 由此就可在不影响正常通信的情况下将 SSL 流量导入隐蔽通信检测服务器.

隐蔽通信检测服务器 DS 对导入的 SSL 流量进行提取, 得到 SSL 通信中服务器所使用的证书, 然后调用基于 iForest 的隐蔽通信检测算法检测此证书是否异常. 若检测结果为证书异常, 即表示当前 SSL 连接为非法隐蔽通信, 则隐蔽通信检测服务器 DS 立刻发送 SSL 连接异常警报事件至 SDN 控制器 Controller. 控制器收到隐蔽通信检测服务器发送的警报后, 立刻对当前网络中与使用非法证书的 SSL 服务器连接的客户端进行溯源, 并下发流表至异常主机挂载的交换机以阻塞此主机发送的 SSL 报文, 从而达到抑制非法隐蔽通信的目的.

2.2 基于 iForest 的隐蔽通信检测算法 (iFCCD)

基于 iForest 的隐蔽通信检测算法 (iFCCD) 对 SSL 报文中提取的服务器证书进行检测以判断网络中是否存在隐蔽通信. 其检测精度受特征值表征证书的准确

度影响, 因此在调用隐蔽通信检测算法之前需提取能够准确表征证书特性的特征值.

在介绍本文提取的证书特征值之前, 引入两个集合[11], 分别是最常见通用名集合 (the most Frequently Appeared CommonName Set, FAS) 和最常见匿名性通用名集合 (the most Frequently Appeared Anonymous CommonName Set, FAAS). 其中 FAS 是根据统计得到的频繁使用的通用名集合, FAAS 也是相同方法得到, 但目前只包含两个元素, 分别为 localhost 和 localdomain.

攻击者制作隐蔽通信使用的证书时, 为保证隐蔽性, 通常采用自签名证书, 并且为了避免自身信息暴露, 此类证书使用者和颁发者的项通常较少. 由于需保持通信的隐蔽性防止被网络安全设备发现, 攻击者在制作证书时通常会设定较短的证书有效时间. 同时攻击者会采用随机构成的域名作为证书通用名, 以此进一步提高隐蔽性. 本文中 FAS 集合和 FAAS 集合就是针对证书的域名而建立的, 若证书的域名在 FAS 中说明其属于网络中常见的域名, 即表示其可信度较高. 若证书的域名出现在 FAAS 中说明其匿名性较强, 即表示此证书的安全性较低.

基于上述特点, 本文采用了 8 个特征值来表征证书, 如表 1 所示, 分别为: (1) 是否为自签名证书; (2) 证书使用者包含的项数; (3) 证书颁发者包含的项数; (4) 证书是否被可信根验证通过; (5) 通用名是否在 FAS 中; (6) 通用名是否在 FAAS 中; (7) 使用者通用名是否符合域名格式 (CN=xxx.com); (8) 证书的有效年份. 上述八个特征值是从证书中提取的关键信息, 能够表征证书的合法性.

表 1 iFCCD 采取的证书特征值

属性描述	特征值	属性及计算细节
自签名证书/自生成证书	1	采用自签名证书/自生成证书时可疑度高
证书使用者包含的项	0.1*项数	通用名, 机构名, 国家名, 地区名; 每项分值为 0.1
证书颁发者包含的项	0.1*项数	通用名, 机构名, 国家名, 地区名; 每项分值为 0.1
证书被可信根验证通过	1	证书链使用可信根, 可疑度低
通用名在 FAS 中	1	流行性强, 可疑度低
通用名在 FAAS 中	1	匿名性强, 可疑度高
使用者的通用名符合域名格式	1	符合格式, 可疑度低
有效期时间	年份	有效期短, 可疑度高

iFCCD 算法采用孤立森林算法 (isolation Forest algorithm, iFOrest), iFOrest 是一种基于决策树的异常检测算法, 其基于数在二叉搜索树中的深度判断数据是否异常, 具有较高的检测精度. 相对于神经网络算法, iFOrest 的训练时间短, 达到相同精度的计算时间少.

本文采用的八个参数特征值对应的当前 iFOrest 中的评价值计算公式如下[15]:

$$S_i = \frac{1}{2} - \frac{1}{2}^{\frac{E(h(x))}{C}} \tag{1}$$

其中, $E(h(x))$ 为特征值 x 在树中的平均深度, C 为当前森林中数的平均深度, 计算公式如下[15]:

$$c(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right) \tag{2}$$

$$H(i) = \ln(i) + 0.577\,215\,6649 \tag{3}$$

隐蔽通信检测启动时, 从配置文件中读取已准备好的训练数据, 训练数据均为从正常证书中提取的相关特征值, 孤立森林算法使用这些特征值训练算法模型. 假设某一特征值训练数据有 L 组, 训练阶段孤立森林有放回的取出 G 组数据, 然后构建 K 棵决策树, 得到 K 棵决策树组成的森林, 并根据孤立森林算法计算出当前森林的阈值 $threshold_i$. 8 个孤立森林分类器均采用上述的训练流程. 孤立森林算法训练完成之后, 隐蔽通信检测正式进入检测阶段.

算法. iFCCD

Input: SSL 流量
Output: 恶意证书检测结果

1. **Begin**
2. Get Hello packet in SSL
3. Extract SSL certificate
4. Calculate features of SSL certificate
5. **if** this certificate is Self-signed certificate **then**
6. append '1' to eigenvalue matrix
7. **else**
8. append '0' to eigenvalue matrix
9. **end if**

```
Append the length of subject to eigenvalue matrix
Append the length of Issuer to eigenvalue matrix
if the root in certificate is trusty then
    append '1' to eigenvalue matrix
else
    append '0' to eigenvalue matrix
end if
if domain name in FAS then
    append '1' to eigenvalue matrix
else
    append '0' to eigenvalue matrix
end if
if domain name in FAAS then
    append '1' to eigenvalue matrix
else
    append '0' to eigenvalue matrix
end if
if subject's domain name is belong to normal format
    then
    append '1' to eigenvalue matrix
else
    append '0' to eigenvalue matrix
end if
Append valid time of certificate to eigenvalue matrix
Send matrix to iForest
if the result of detection is true then
    send result to controller
else
    goto end
end if
End
```

进行检测时, 隐蔽通信检测算法接收新的证书特征值矩阵, 然后调用训练完成的算法模型判断此证书特征值是否为异常. 当待检测的特征值矩阵输入时, 各特征值分别送入相应的分类器. 在各自的分类器中, 每个决策树返回待测特征值在树中的层数, 然后得到当前特征值在该森林中的评价值 S_i. 在一个分类器中, 若特征值评价值小于该分类器阈值, 则表示当前数据为异常数据.

在 iFCCD 中, 最后计算分类器的评价值, 若该评价值小于阈值则判定当前特征值矩阵为异常, 即当前证书为非法证书; 否则判定为正常证书, 如公式 (4) 所示:

$$\sum_{8}^{i=1} S_i < \sum_{8}^{i=1} threshold_i \tag{4}$$

综上所述, 为准确表征证书特征, iFCCD 算法采用八个特征值表征证书信息, 同时使用孤立森林作为异常检测算法, 提高非法证书的检测精度. 在训练阶段中, iForest 的训练数据均为正常证书的特征值. 在检测阶段中, iFCCD 利用已训练的 iForest 模型分别对 8 个特征进行判定, 最后综合判定表征证书信息的八元组矩阵是否异常, 以此可判定网络中是否存在隐蔽通信.

3 实验验证与分析

3.1 实验环境

本文采用的数据集有三个: ITOC2009[16]、Contagio Malware Dump (CMD)[17]以及从本地抓取的正常 HTTPS 证书. 训练数据中有 100 个证书, 其中 50 个为 ITOC 数据集中的正常证书, 50 个为 CMD 数据集中的正常证书. 测试数据有 271 个证书, 其中包括 74 个 CMD 数据集中正常的证书, 26 个本地抓取的正常证书以及 171 个 CMD 数据集中的非法证书. 首先提取训练数据集中证书的特征值, 然后将训练证书特征值作为输入训练孤立森林算法. 在训练 iForest 算法模型时, 从 100 组训练数据中有放回地取出 75 组数据, 构建 100 棵决策树.

本文采用文献[11]中提到的证书可信度计算算法 (文献[11]中将其简称为 CCD 算法) 作为对比实验, 其中计算结果小于 0 时视为证书非常可疑, 计算结果大于 0 小于 0.85 时视为证书较可疑, 大于 0.85 则认为证书正常. 本文使用受试者工作特征曲线 (Receiver Operating Characteristic curve, ROC 曲线) 衡量 iFCCD 与 CCD 两种算法的检测精度.

3.2 实验结果

本文对上述提出的检测算法的实验验证过程如下: 首先使用训练数据训练 iForest 模型, 检测时将数据集里的证书依次提取出来, 获得其中的证书特征值, 然后将证书特征值依次送入训练好的 iForest 模型中. 在衡量 iFCCD 与 CCD 两种算法精度时, 通过动态调整 iForest 的阈值得到的 ROC 曲线如图 3 所示.

图 3 中, 横坐标为误检率, 纵坐标为检测精度, 其中实线为本文提出的 iFCCD 算法得到的检测结果, 虚线为对比算法 CCD 的检测结果. 从图中可以看到, iFCCD 算法的误检率在 16% 时可以达到 100% 的检测精度, 而 CCD 算法要达到 100% 的检测精度其误检率需达到 24%. 显然本文提出的证书检测算法在提高证书检测精度的同时能够降低误检率. 图 3 中, 两种算法的 ROC 曲线均在误检率达到一定值时快速上升, 出现上述现象的原因是数据集里的非法证书的特征值基本

相似.

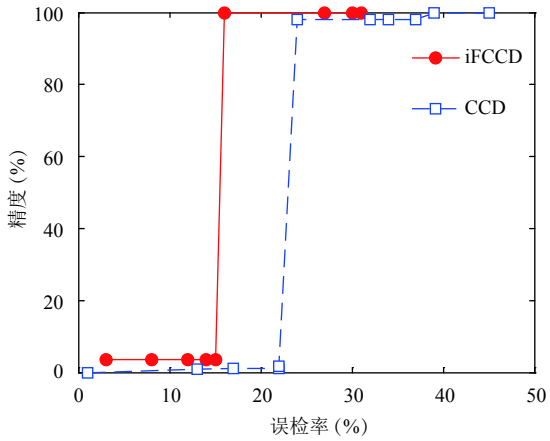

图 3 iFCCD 及 CCD 算法 ROC 曲线图

3.3 实验结果分析

为进一步说明 iFCCD 及 CCD 算法的检测精度, 本节对 iFCCD 及 CCD 算法的检测结果进行了详细分析.

数据集 ITOC2009 中出现了大量类似如下格式的非法证书: “Version=3; Issure:C=--; ST=SomeState; Validity:one year; OU=SomeOriganizationalUnit; EMAIL=root@localhost.localdomain; O=SomeOrganization; L=SomeCity;”, 对于上述非法证书, CCD 算法计算出的可信度为–0.1, CCD 算法认定其为非常可疑的证书, 既 CCD 算法能够成功检测上述非法证书.

但是对于图 4 所示的非法证书, 使用 CCD 算法计算的可信度为 0.35(属于较可疑级别), 其不能精确检测上述非法证书. 通过进一步分析可知此证书属于非法证书, 因为其使用者项中都是匿名信息符合非法证书的特征, 而且证书有效时间较短. 此证书相对于 ITOC2009 数据集的一般非法证书仅仅将使用者 CN 换了一个匿名, 就使得 CCD 产生漏检. 虽然可以将匿名域名加入到 FAAS 中提高 CCD 的检测精度, 但匿名性域名范围太广无法将全部新匿名域名包含在集合中, 因此采用 CCD 方法容易导致漏检. 而在采用 iFCCD 方法检测上述非法证书时, 由于 iForest 所具有的检测精度高的优点, 所以 iFCCD 方法能够将其判定为非法证书.

与 CCD 算法相比, iFCCD 算法的误检率更低. 图 5 所示为 ITOC2009 的合法证书, CCD 算法计算该证书可信度为-0.1, 判定其为非法证书, 但是经查询得知此证书为根级证书属于可信任证书, CCD 算法明显导致了误检测. 而 iFCCD 算法能够正确判定其为合法证书.

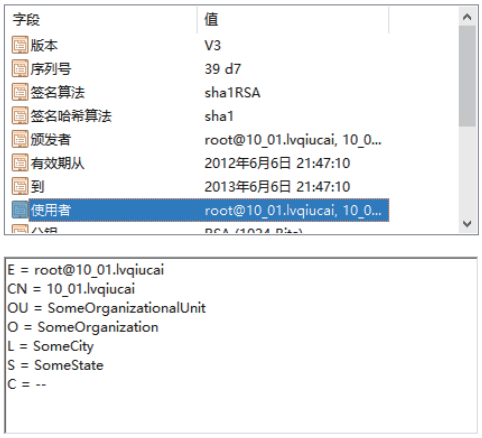

图 4 非法证书使用者信息示例

图 5 正常证书使用者信息示例

从上述分析得知, 本文提出的基于 iForest 的隐蔽通信检测机制能够在降低误检率的同时提高检测精度.

4 总结

为解决 SDN 网络中存在的隐蔽通信检测问题, 本文提出了面向 SDN 的隐蔽通信检测机制, 该机制利用 SDN 的特性准确获取网络中可能存在的隐蔽通信流量. 针对 CCD 方法较为模糊的判定结果问题, 本文提出了基于 iForest 算法的隐蔽通信检测算法 iFCCD, 该算法可以降低误检率、提高检测精度, 同时避免了使用经验值作为阈值可能导致的误检率增高或精度降低的问题. 本文提出的 iFCCD 方法具有较好的可扩展性,

只需改变提取证书的特征值或训练数据集就可运用于不同场景的 SDN 网络.

参考文献

1 Nunes BAA, Mendonca M, Nguyen XN, *et al*. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials, 2014, 16(3): 1617–1634. [doi: 10.1109/SURV.2014.012214.00180]

2 McKeown N, Anderson T, Balakrishnan H, *et al*. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69–74. [doi: 10.1145/1355734]

3 Casado M, Freedman MJ, Pettit J, *et al*. Ethane: Taking control of the enterprise. ACM SIGCOMM Computer Communication Review. Kyoto, Japan. 2007. 1–12. [doi: 10.1145/1282380.1282382]

4 Yoon MS, Kamal AE. Power minimization in fat-tree SDN datacenter operation. Proceeding of 2015 IEEE Global Communications Conference. San Diego, CA, USA. 2015. 1–7. [doi: 10.1109/GLOCOM.2014.7417135]

5 Daly MK. Advanced persistent threat. USENIX 23rd Large Installation System Administration Conference. Baltimore, MD, USA. 2009, 4(4): 2013–2016.

6 Tankard C. Advanced persistent threats and how to monitor and deter them. Network Security, 2011, 2011(8): 16–19. [doi: 10.1016/S1353-4858(11)70086-1]

7 Li MC, Huang W, Wang YB, *et al*. The study of APT attack stage model. Proceedings of 2016 IEEE/ACIS 15th International Conference on Computer and Information Science. Okayama, Japan. 2016. 1–5. [doi: 10.1109/ICIS.2016.7550947]

8 Chandra JV, Challa N, Pasupuletti SK. A defense approach from advanced persistent threat through defense in depth mechanism for cloud security. Advanced Science and Technology Letters, 2017, 147: 268–275.

9 Rass S, König S, Schauer S. Defending against advanced persistent threats using game-theory. PLoS One, 2017, 12(1): e0168675. [doi: 10.1371/journal.pone.0168675]

10 Sood AK, Enbody RJ. Targeted cyberattacks: A superset of advanced persistent threats. IEEE Security & Privacy, 2013, 11(1): 54–61.

11 Cao ZG, Xiong G, Zhao Y, *et al*. Two-phased method for detecting evasive network attack channels. China Communications, 2014, 11(8): 47–58. [doi: 10.1109/CC.2014.6911087]

12 Fu PP, Guo L, Xiong G, *et al*. Classification research on SSL encrypted application. In: Yuan Y, Wu X, Lu Y, eds. Trustworthy Computing and Services. ISCTCS 2012. Communications in Computer and Information Science, vol 320. Springer, Berlin, Heidelberg. 2012. 404–411. [doi: 10.1007/978-3-642-35795-4_51]

13 Bro-Project. 2017. Intelligence Framework. https://www.bro.org/sphinx/frameworks/intel.html. [2018-05-21].

14 Ghafir I, Přenosil V, Hammoudeh M, *et al*. Malicious SSL certificate detection: A step towards advanced persistent threat defence. Proceedings of International Conference on Future Networks and Distributed Systems. Cambridge, Britain. 2017. 1–6. [doi: 10.1145/3102304.3102331]

15 Liu FT, Ting KM, Zhou ZH. Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, 6(1): 1–39. [doi: 10.13140/RG.2.1.2591.8165]

16 ITOC research: CDX datasets. http://www.oalib.com/references/13245491.

17 Contagio Malware Dump: Collection of PCAP files categorized as APT, Crime or Metasplot. http://contagiodump.blogspot.com/.