

基于无向图流程创建聚合推理的 k 近邻 SDN 可疑流量检测

徐 镇, 张小平, 张海翔

(清华大学 计算机科学与技术系, 北京 100084)

摘 要: 为提高软件定义网络(Software Defined Network, SDN)可疑流量检测算法性能, 提出一种基于无向图流程创建信息聚合推理的 k 近邻 SDN 可疑流量检测方法. 利用 OpenFlow 模块进行数据流创建, 并进行入侵规则构建. 然后, 基于马尔可夫链的图节点/边表示, 采用无向图方式进行攻击特征表示, 实现新增攻击的增量化表示, 降低无向图构建的计算复杂度, 并利用 k 近邻算法对无向图恶意攻击流量特征进行分类, 实现攻击有效检测. 最后, 通过构建的 SDN 测试平台, 对所提算法的性能进行验证.

关键词: 软件定义网络; 无向图; 信息聚合; 推理系统; k 近邻; 可疑流量检测

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2017)12-0005-06

Undirected Graph Information Aggregation Reasoning Based SDN Suspicious Traffic Detection

XU Zhen, ZHANG Xiao-ping, ZHANG Hai-xiang

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: In order to improve the performance of suspicious traffic detection algorithm in software defined network, this paper proposes a method for detecting suspicious traffic of k nearest neighbor based on undirected graph process. OpenFlow module is used to create data stream, and the intrusion rules are constructed. Then, based on the map nodes/edges represent with the Markov chain, the undirected graph was used to represent the attack characteristics, it realized the incremental implementation of new attacks, which could reduce the computational complexity of constructing undirected graph, and the k nearest neighbor algorithm was used to classify the malicious attack traffic characteristics of undirected graph to achieve effective detection of attacks. Finally, the performance of the proposed algorithm is verified by the SDN test platform.

Key words: software defined network; undirected graph; information aggregation; inference system; k nearest neighbor; suspicious traffic detection

1 引言

在网络安全中, 自主网络 agents 可通过对网络拓扑结构和入侵数据特性学习, 实现网络规则更新, 实现最小人力投入情况下对入侵数据的有效检测. 在可编程网络如软件定义网络可能需要较长时间的构建网络 agents, 从而实现网络完全自治^[1-2].

尽管如此, 专业学者认为 SDNs 是非常脆弱的, 容易成为攻击目标^[3]. 当转发行为的逻辑行为集中

和分配在控制器中, 则构建了一个单一的故障点和攻击点. 无论是利用控制器中的漏洞或交换机和控制器之间的通信链路, 都可能带来一些攻击行为, 如拒绝服务(DoS)和主机位置劫持攻击等, 需要高度关注 SDNs 安全问题^[4]. 但是 SDNs 安全研究存在一定挑战, 首先, 有必要创建更为复杂的安全措施, 不只是采取丢弃或转发流量方式. 具体而言, 统计和签名匹配技术需要结合数据流规则生成. 其次, 用于创建安全应用程序的现有程序依赖于管理网络的控

制器的体系结构. 第三, 由于 SDNs 流量处理的动态性, 使用异常检测技术很难发现不同类型的攻击, 会导致很高的误报率.

为解决这些问题, 本文提出一种新的入侵检测算法可实现对 SDNs 网络的有效识别. 所提算法利用了 OpenFlow 模型、基于马尔可夫链的图节点/边表示模型、k 近邻查询算法. 本文主要创新点:

(1) 证明了使用现有的攻击特征来识别 SDNs 攻击的可行性.

(2) 创建了一个灵活的方法, 可使用包含标记的事件在不同层面上构建攻击预测模型, 以确定 SDNs 攻击数据集.

(3) 证明了该方法通过在运行时查询预创建的图形, 可有效地实现 SDNs 的攻击识别. 实验结果表明所提方法优于传统的分类器和现有 SDNs 攻击分类技术.

(4) 提出了一个图模型的增量创建技术, 通过添加新类型的节点使其与现有节点建立关联.

2 系统模型

考虑如图 1 所示 SDNs, 基于 SDN 的网络架构由一个 SDN 控制器和 OF 启用开关构成. OF 是一种通信网络协议, 使 SDN 控制器可访问交换机的转发表. 控制平面上, SDN 控制器与 OF 启用开关直接连接, 通过这种连接可在每个交换机上进行转发表流量控制和采集. SDN 控制器通过启用开关可实现对所有数据包的交换操作^[5-6].

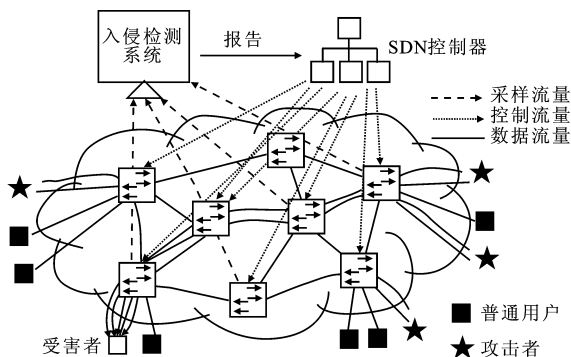


图 1 软件定义网络

图 1 显示了一个基于 SDN 网络的 IDS 对可疑流量检查. 对于基于 SDN 的 IDS 流量监测, 可通过镜像方法再任何启用开关进行数据包采样, 实现 SDN 的完全可配置. IDS 对交换机所有镜像数据包进行检查, 如果检测到网络流量可疑数据包, 则产生警报信息, 反馈至 SDN 控制器. 基于交换机状态和

IDS 检查结果, SDN 控制器可重新配置网络, 以抵御网络攻击.

假设有 f 个流量包和 N 个网络启用开关. 令 λ_i 表示属于 i 数据流的恶意攻击率. 如果数据流不包含任何攻击数据包, 其恶意攻击率是 0. 数据流的发送速率可由发送速率向量表示: $s = [s_1, \dots, s_f]^T$, 恶意流量可表示为: $\lambda = [\lambda_1, \dots, \lambda_f]^T$. 恶意速率的发送单位是包/秒.

3 基于推理的网络可疑流量检测

3.1 攻击预测系统框架

利用定期流量标记对不同的样品进行分析. 所提方法目标定义为: (1) 检查是否存在攻击签名可以用来分析 SDNs 所产生的痕迹. (2) 研究传统可疑流量检测系统如何处理 SDNs 所产生的大量痕迹. 图 2 所示为本文所提攻击预测系统框架^[7-8].

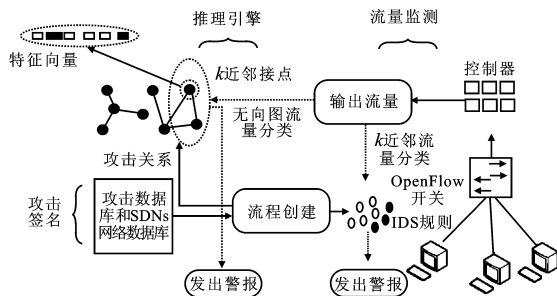


图 2 攻击预测系统框架

图 2 中, 流量监测模块 (Traffic monitor) 该根据网络或交换机的要求, 通过 OpenFlow 交换机进行定制流量接收和定期的流量查询. 然后, 可利用攻击预测模型对数据流记录进行分析和导出; 攻击特征提取模块 (Attack signatures) 用于表示从网络数据集中提取的攻击签名. 对存储在网络攻击数据库内的数据进行融合, 利用流创建模块 (flow creation) 生成 NetFlows.

3.2 基于 OpenFlow 流创建模块

需要软硬件配合实现在数据包轨迹中进行数据流的收集和输出. 数据流包括特定功能数据包共享, 例如产生时间、数据源和目标 IP^[9]. 在本文算法中涉及到数据包的十个特征: 目标 IP 和数据源、目标端口和数据源端口, 字节和数据包数, 结束和开始时间, TCP 标志和流动时间. 在算法 1 中给出聚合数据包流量创建步骤.

算法 1: 基于 OpenFlow 信息聚合的流创建

输入: 数据流持续时间 duration, 默认 30 秒; 特征 features: 从数据包中提取特征列表; 流量包 packets, OpenFlow 网络的痕迹;

```
1. procedure createFlow(duration, features, packets)
2. 初始化数据包列表  $S=\emptyset$ ;
3. for all packet do
4. if  $S=\emptyset$  then packet $\rightarrow S$  else
5.     if packet 开始时间与 S 间持续时间大于 duration do
6. 输出第一数据流, 并从 M 中读取第一数据流;
7. 从 S 中移除首个数据流;
8. 从 M 中移除首个数据流;
9.     end if
10.    初始化 flag=False;
11. for packet $\in S$  do
12. if featurepacket = featurestartpacket then
13. M. get (id of start Packet). add (packet);
19. 设置 flag=True;
20.     end if
21.     end for
22.     if flag=False then
23.         packet $\rightarrow S$ ;
24. M. put (id of packet, add packet to the list);
25.     end if
26. end if
27. end for
28. if  $M\neq\emptyset$  then
29. 输出 M 中所有数据流;
30. end if
31. end procedure
```

该过程是预处理步骤,其仅在存储在攻击数据库中的数据不是流数据时执行. 使用持续时间参数调整每个流的存在时间.

3.3 可疑流量检测规则

IDS 规则通过特定机器学习技术或网络管理员或进行访问或更新. 我们的方法是对基于 Open-Flow 的数据流进行分析,以决定是否存在可疑的攻击行为. 这里使用流分类模型来创建这套规则. 这些规则是使用存储在攻击数据库中的常规流来创建的. 令 $\bar{V}=[f_1:d_1,\cdots,f_n:d_n]$ 是从网络跟踪中提取的一组特征,见图 2 所示. 假设流量分类模型 m 检测 $\bar{V}$. 如果 $\bar{V}$ 同 m 定义的任何签名相匹配,则发出警报. 假定 $N=\{n_1,n_2,\cdots,n_p\}$ 是由 m 确定的所有可能警报的集合. 假设 NS 是事先存在的可疑流量的标签集合, NB 是事先存在的良性流动的标签集合,所使用的任何流分类模型,在运行时需满足两个目标:(1) 在给定 NS 集合正确标签情况下,提高可疑活动的检出率. (2) 在给定 NB 集合正确标签情况下,降低良性流动误报率.

3.4 节点路径推理

使用特定的特征对每个节点 n_i 进行表示(即相应的数据流特征). 在收集的数据中的每个标签被视为图中的节点. 利用单一特征向量 V_{n_i} 来表示每个节点 n_i ,如表 1 所示.

表 1 节点间相关特征向量

特征向量	f_{1_b1}	f_{1_b2}	f_{2_b1}	f_{2_b2}	f_{n_b1}	f_{n_bj}
V_{n_i}	0.91	0.66	0.37	0.44	0.54	0.64
V_{n_j}	0.82	0.42	0.55	0.41	0.42	0.64

每个向量包含特征源和目标 IP、端口号、数据包数、TCP 标志、协议类型和警报描述的聚合权重. 每个特征 f_{i_bj} 的归一化频率将作为其在节点处的权重. 因为这些特征都以数值化形式表示,可使用离散的宽度相等的分级方法,将特征 f_i 表示为 $b_1,b_2,\cdots,b_j$ 形式.

使用皮尔森相关进行节点间的相关性计算,如下所示.

$$\text{sim}_{n_i,n_j} = \frac{(\text{cov}(\sigma_{V_{n_i}},\sigma_{V_{n_j}}))}{\sigma_{V_{n_i}} \times \sigma_{V_{n_j}}} e((V_{n_i} - \mu_{V_{n_i}})(V_{n_j} - \mu_{V_{n_j}})) \quad (1)$$

将所得的权重进行归一化,然后分配连接节点的边缘上.

在节点间间接路径推理发现过程中,在基于遍历路径长度基础上,进行相似性图形创建. 选择数值得分最高的作为最可行的路径,表示任何两个节点之间的关系可能性. 图 3 显示了一个具有四个节点的图例.

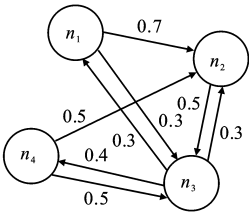


图 3 基于马尔可夫链的图节点 / 边表示.

图中边缘上显示的数值表示从一个节点到另一个节点的跃迁概率. 可根据下列步骤确定最可能的关系:

Step1: 利用连接节点 (n_i,n_j) 所有路径的权重和长度 $|t|$ 的乘积之和,进行分数 l 计算.

Step2: 根据节点 (n_i,n_j) 长度为 $|t|$ 的所有路径数量, $l = N_{n_i-n_j}^{|t|}$, 则不同的 $|t|$ 取值,会有不同的分数 l 取值,选取最大值作为最有可能的节点链接.

当发现了新节点 n_i , 没有必要重新生成整个图

形. 可采用以下步骤进行简化:

Step1: 查找新节点 n_i 与现有图中所有其他节点之间的相似度.

Step2: 根据相似度值, 利用 k 近邻算法在已有图像中选取与新节点 n_i 最为接近的节点.

Step3: 计算上述 k 近邻算法选取的 k 个节点的分数 l 平均值, l_{avg} .

Step4: 如果所有选取的 k 个节点属于 c_i , 即所有的节点属于同一攻击类型.

$$l_{\langle n_i \rightarrow n_j \rangle} = l_{\text{avg}} + (|1 - l_{\text{avg}}| \times \text{sim}(n_i, n_j)) \quad (2)$$

如果所有选取的 k 个节点不属于 c_i , 则可计算:

$$l_{\langle n_i \rightarrow n_j \rangle} = l_{\text{avg}} - (|1 - l_{\text{avg}}| \times \text{sim}(n_i, n_j)) \quad (3)$$

3.5 无向图模型 k 近邻流量分类

k 近邻分类器^[10], 采用了相似函数对 k 个近似节点进行发现, 图 4 给出每个节点利用特征向量表示的实例.

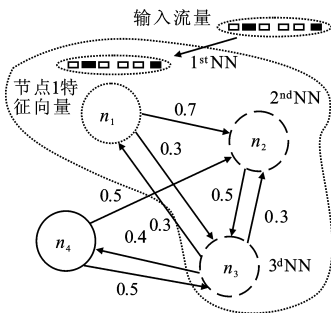


图 4 近似节点发现

每个特征向量代表一个配置文件, 其中包含该节点的最高权重特征. 利用条件熵进行特征权重赋值, 以创建配置文件. 数据流根据节点邻域内主要节点类型进行分类, 例如若 $k = 1$, 则数据流根据单一节点类型进行分类. 每个数据流被归类为特定类型的 DoS 攻击或良性活动. 利用无向图模型进行数据流分类, 具体见算法 2 步骤所示.

算法 2: 无向图模型数据流分类

输入: 数据流集合 $FL = \{fl_1, fl_2, \dots, fl_n\}, l(n_i, n_j), \forall n_i, n_j \in N$

1. Procedure Flow classification(FL, k, l);
2. for $i = 1:n$ do
3. $R = \{\emptyset\}$, suspiciouscount = 0, benigncount = 0;
4. 基于特征相似查找与 fl_i 最为相似的节点 n_i ; $R = \{n_i\}$;
5. if n_i 是可疑节点 then
- suspiciouscount = suspiciouscount + 1;
6. else benigncount = benigncount + 1;
7. end if
8. for $j = 1:k$ do

9. 检索 top $kl(n_i, n_j)$; $R = \{R\} \cup n_j$;

10. if n_j 为 DoS 警报 then

11. suspiciouscount = suspiciouscount + 1;

12. else

13. benigncount = benigncount + 1;

14. endif

15. endfor

16. classifyF(R, fl_i);

17. endfor

18. endProcedure

Function classifyF(R, fl_i)

1. 输入: 带有可疑和 / 或良性节点的集合 R , 数据流 fl_i ;

2. 返回: 每个数据流 fl_i 的类型;

3. 检查 R 中的良性和可疑节点计数;

4. if benigncount > suspiciouscount then

5. 返回“良性数据流”;

6. elseif benigncount < suspiciouscount then

7. 返回“Dos 攻击数据流”;

8. else 从 R 中删除 k 近邻节点; $R = (R \setminus k)$;

9. classifyF(R, fl_i)

10. endloop

11. endFunction

该算法需要三个输入, 需要处理的数据流集 FL , 最近邻居数量 k 和分数值 l . 利用数据流特征进行图中 k 近邻节点选取. 在图中的每个节点表示为一个特征向量, 用来发现传入数据流的相似特性(第 3 行). 与 k 近邻分类器相比, 一旦输入数据流 fl_i 的最接近节点 n_i 被发现, 则利用分数值 l 对节点 n_i 的 k 近邻节点进行查询(第 8 行).

4 实验分析

4.1 实验平台

为验证算法有效性, 构建了一个 SDN 测试平台, 并在该测试平台上实现了算法性能测试. 该实验台由 HP Z420 工作站构成, 配备英特尔至强 e5-1620 处理器, 运行环境是 Ubuntu 14.04.2 LTS, 其作为 SDN 控制器, 6 Odroid-xu3 嵌入式主板作为 SDN 交换机作为 SDN 开关, 戴尔 OptiPlex 台式电脑作为 IDS, 拓扑结构见图 5 所示. 测试平台具体细节如下:

(1) SDN 控制器: 存在多种 SDN 控制器实现方式, 如 FloodLight、ONOS 和 OpenDaylight(ODL). 其中, 我们选择了 ODL 作为测试平台的 SDN 控制器. ODL 是 Java 编写的开源项目软件. 其支持 OpenFlow 协议, 并提供编程接口的北桥 API (REST API) 和南桥 API (OSGi).

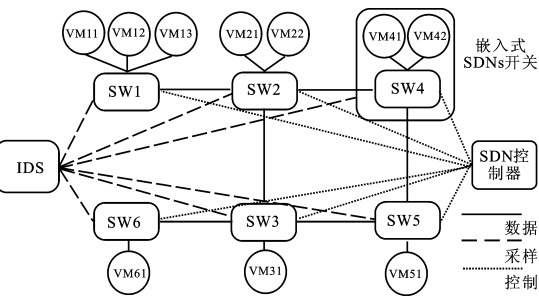


图 5 SDN 测试平台拓扑结构

(2)SDN 交换机和流量源:构建了 SDN 与 SDN 启用开关. Odroid-xu3 嵌入板安装有 Open vSwitch (OVS)软件作为 SDN 开关. 由于每个板只配备了单一以太网端口和 USB 集线器,这里增加 USB 到以太网适配器获得四个附加以太网端口. 一个连接到 SDN 控制器另一个用于镜像流量采样,其他用于构建数据网络拓扑结构.

(3)IDS 配置:Snort 和 Suricata 是最流行的开

源可疑流量检测系统. 在测试中,利用 Snort 检测 VMs 所产生的恶意流量. 攻击者假定使用端口号 10 000 执行攻击端口扫描.

4.2 结果分析

实验中采用的数据库有两种:(1)包含 DOS 攻击的 CAIDA 数据集,该数据集包含匿名的网络痕迹. 其所包含的攻击,主要以消耗计算资源为目的,以阻止用户对服务器的访问. 此数据集没有每个数据包的标签,因此,对该数据集执行 Snort IDS 分析,以确定数据包类型. (2)包含可疑网络连接的数据集,该数据集包含有数据源和目标 IP、数据源和目标端口、协议和分组类型特征的网络连接数据包.

对每个数据集产生的流量执行预处理步骤. 通过改变流量参数,为测试过程选择不同类型的攻击和良性活动. 实验中使用的节点数量和类型如表 2 所示.

表 2 实验测试数据

类型	数据集 1			数据集 2		
	训练样本数	测试样本数	节点数量 n	训练样本数	测试样本数	节点数量 n
TCP/SYN Flood	6 213	3 217	1	8 315	1 215	1
UDP Flood	—	2 513	—	5 321	—	4
ICMP Flood	6 231	3 589	2	7 256	2 613	2
Total	12 444	9 319	—	20 892	3 828	—
TCP/SYN Benign traffic	2 713	1 102	1	4 511	3 652	1
UDP Benign traffic	—	—	—	6 623	—	4
ICMP Benign traffic	5 113	4 102	2	7 214	2 213	2
Total	7 826	5 204	—	18 348	5 865	—
Suspicious and benign	20 267	14 516	—	39 241	9 687	—
suspicious flows 比例	0.62	0.63	—	0.45	0.62	—
benign flows 比例	0.37	0.34	—	0.52	0.38	—

实验 1:(参数 k 性能影响) 本文算法中部分用到了 k 近邻算法,参数 k 对于算法产生较大影响. 这里选取标准 k 近邻算法对参数 $k(k = 1 \sim 6)$ 的不同取值对检测性能影响进行对比分析. 对比指标选取检测精度、召回率和 F-score 三组指标,实验结果如图 6 所示.

根据图 6 结果可知,参数 k 对于算法会产生较为明显的影响,但是其取值并不是越大或越小越好,而是在 $k = 3$ 左右会取得最佳效果,可看出在两种数据集上均呈现出 $k = 3$ 取值的最佳性,这说明 $k = 3$ 具有一定的普适性. 同时,可看出在各种参数 k 取值情况下,本文算法的性能指标值均要优于选取的标准 k 近邻算法.

实验 2:(参数 k 计算效率影响) 选取标准 k 近邻算法对参数 $k(k = 1 \sim 6)$ 的不同取值对计算效率影响进行对比分析. 对比指标选取算法运行时间,实验结果见图 7 所示.

根据图 7 所示结果,参数 k 对算法计算效率的影响较为明显,并且呈现明显的单调递增特点,参数 k 的取值越大,算法的计算时间越长,因此结合图 6 所示结果,选取参数 $k = 3$ 取值是较为合适的做法. 同时可看出,本文算法在计算效率上要较为明显的优于标准 k 近邻算法.

实验 3:(算法综合性能对比) 为更充分验证算法有效性,对比算法选取 SVM 算法、SOM 算法、SVM & PCA 算法. 对比指标选取检测精度(P)、召

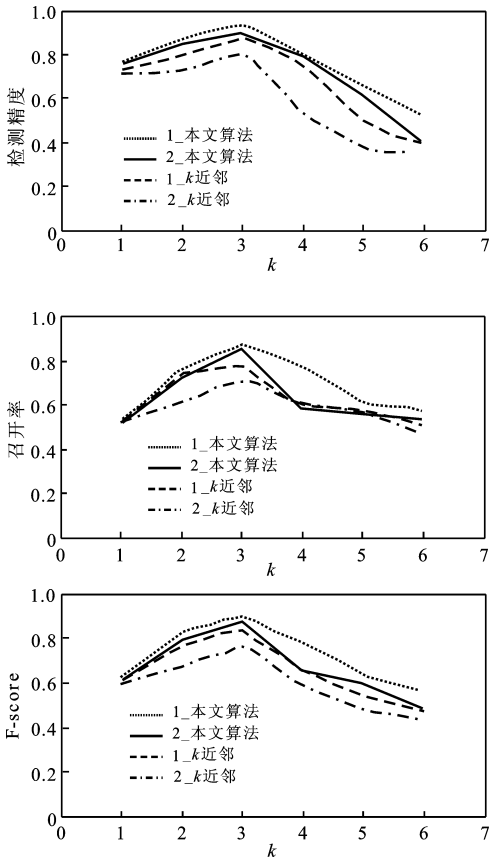


图 6 参数 k 检测性能影响

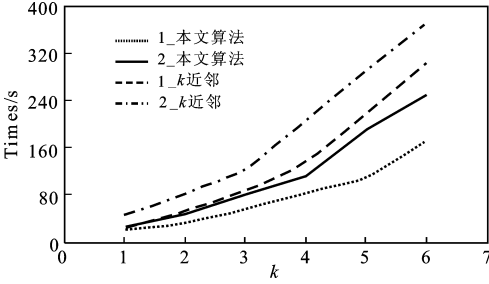


图 7 参数 k 计算效率影响

回率(R)和 F -score(F) 三组指标,数据集选取上述两组数据集,实验结果见表 3 所示。

表 3 算法综合性能对比

对比算法	数据集 1			数据集 2		
	P	R	F	P	R	F
本文算法	0.91	0.86	0.88	0.87	0.83	0.86
SOM	0.72	0.68	0.69	0.76	0.75	0.76
SVM&PCA	0.77	0.73	0.75	0.82	0.76	0.78
SVM	0.78	0.65	0.73	0.76	0.75	0.78

根据表 3 实验结果可知,在对比算法指标上,本文算法要明显优于选取的 SVM 算法、SOM 算法和 SVM & PCA 算法,SVM & PCA 算法的检测性能要优于 SVM 算法和 SOM 算法,而 SVM 算法和 SOM 算法检测性能相差不大,互有优势,上述结果

验证了所提算法较高的恶意攻击检测能力。

5 结束语

本文提出一种基于无向图流程创建信息聚合推理的 k 近邻 SDN 可疑流量检测方法,利用 OpenFlow 模块进行数据流创建,并进行入侵规则构建,采用无向图方式进行攻击特征表示,实现新增攻击的增量化表示,降低无向图构建的计算复杂度,并利用 k 近邻算法对无向图恶意攻击流量特征进行分类。今后,这项工作将扩展到考虑其他类型攻击的 SDNs,同时预测模型的 OpenFlow 应用嵌入开发也是进化研究方向。

参考文献:

[1] 魏振伟,刘飞. 粒子群特征优选的 SVDD 入侵检测研究[J]. 微电子学与计算机, 2016, 33(8): 144-148.

[2] Fukun Bi, Tong Zheng E, Hongquan Qu, et al. A harmful-intrusion detection method based on background reconstruction and two-dimensional K-S test in an optical fiber pre-warning system[J]. Photonic Sensors, 2016, 6(2): 143-152.

[3] 谢修娟,莫凌飞,朱林. 基于关联规则的滥用人侵检测系统的研究与实现[J]. 现代电子技术, 2017, 40(2): 43-47.

[4] Muhammad Syafiq Mohd Pozi, Md Nasir Sulaiman, Norwati Mustapha, et al. Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming [J]. Neural Processing Letters, 2016, 44(2): 279-290.

[5] Takashi Onoda. Probabilistic models-based intrusion detection using sequence characteristics in control system communication[J]. Neural Computing and Applications, 2016, 27(5): 1119-1127.

[6] Mazhar M Rathore, Awais Ahmad, Anand Paul. Real time intrusion detection system for ultra-high-speed big data environments[J]. The Journal of Supercomputing, 2016, 72(9): 3489-3510.

[7] Pandeewari N, Ganesh Kumar. Anomaly detection system in cloud environment using fuzzy clustering based ANN[J]. Mobile Networks and Applications, 2016, 21(3): 494-505.

[8] Dixiu Xue, Rong Zhang, Hui Feng, et al. CNN-SVM for microvascular morphological type recognition with data augmentation[J]. Journal of Medical and Biological Engineering, 2016, 36(6): 755-764.

(6): 2651-2662.

[3] Knaus C, Zwicker M. Dual-domain image denoising [C]//Image Processing (ICIP), 2013 20th IEEE International Conference on, Melbourne, Australia, IEEE, 2013: 440-444.

[4] 肖进胜,李文昊,姜红,等. 基于双域滤波的三维块匹配视频去噪算法[J]. 通信学报,2015,35(9):91-97.

[5] Hwang H, Haddad R A. Adaptive median filters: new algorithms and results[J]. IEEE Transactions on image processing, 1995, 4(4): 499-502.

[6] Cai J F, Chan R H, Nikolova M. Fast two-phase image deblurring under impulse noise[J]. Journal of Mathematical Imaging and Vision, 2010, 36 (1): 46-53.

[7] Mitra K, Veeraraghavan A, Chellappa R. Robust rvm regression using sparse outlier model[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. San Francisco, CA, USA, IEEE, 2010: 1887-1894.

[8] Liu J, Tai X C, Huang H, et al. A weighted dictionary learning model for denoising images corrupted by mixed noise[J]. IEEE transactions on image processing, 2013, 22(3): 1108-1120.

[9] Aggarwal H K, Majumdar A. Hyperspectral image denoising using spatio-spectral total variation [J]. IEEE Geoscience and Remote Sensing Letters, 2016, 13(3): 442-446.

[10] Dong C, Loy C C, Tang X. Accelerating the super-resolution convolutional neural network[C]//European Conference on Computer Vision. China, Hong Kong, Springer International Publishing, 2016: 391-407.

作者简介:

雷俊锋 男,(1975-),博士,副教授. 研究方向为图像处理. E-mail:jflei@whu.edu.cn.

王 赫 男,(1993-),硕士. 研究方向为图像处理.

刘恩雨 女,(1994-),硕士. 研究方向为深度学习.

肖进胜 男,(1975-),博士,副教授. 研究方向为深度学习、视频图像处理.

谢文娟 女,(1980-),硕士. 研究方向为信息通信技术.

(上接第 10 页)

[9] Yao Li, Lingqiao Liu, Chunhua Shen. Mining mid-level visual patterns with deep cnn activations[J]. International Journal of Computer Vision, 2017, 121(3): 344-364.

[10] 汪伦杰,廖兴宇,潘伟杰. 基于信号均值滤波+k-means+WKNN 的 Wifi 指纹定位算法研究[J]. 微电子学与计算机, 2017, 34(3): 30-34.

作者简介:

徐 镇 男,(1987-),硕士研究生. 研究方向为网络安全、入侵检测. E-mail:xuzhenhua@163.com.

张小平 男,(1977-),博士,副教授. 研究方向为网络协议、物联网、网络全.

张海翔 男,(1990-),硕士研究生. 研究方向为网络安全、物联网.