Real-time identification of encrypted VOIP application using deep learning approach

**KeyWords** deep learning; bi-flow; VOIP; traffic identification

**Abstract**

VoIP凭借其服务质量以及价格优势获得了越来越多用户的青睐。但因为VoIP服务的便利性也造成了一些社会悲剧。为了使VoIP应用更好的服务于人类，对恶意VoIP的监管是重要的。但是VoIP技术的升级使得传统的识别方法效率低下。

因此，文章针对加密的VoIP应用提出了一种使用deep learning提取语音流量特征的方法，并使用所获得的特征进行voip应用识别。对比现存的VoIP应用的识别方法，本文所提出的方法致力于在Voip通话开始阶段对其进行识别，不依赖于整个通话过程所产生的流量。其次，本文提出的提取特征的方法较人为的提取方法更为可靠。

**1. Introduction**

**Recnetly,** VoIP（Voice over Internet Protocol）应用随着其服务质量的提高凭借其优越的价格优势，在广大人群中的得到了越来越多的使用。Voip应用的广泛使用使网络监管的难度大大提高。对voip应用流量进行有效的识别可以帮助解决网络监管问题。

大多数的VOIP应用使用SIP协议或者H.323协议进行通话连接，使用RTP协议进行通话数据传输。随着voip技术的升级，大多数的voip应用会对连接过程和通话传输过程进行加密。在通信建立阶段可能会使用SSL/TLS，SIPS, WEP,WAP/WAP2等加密技术进行加密，在语音传输阶段可能会使用SSL/TLS，SRTP/SRTCP等加密技术进行加密。VOIP服务的完善使得基于端口、特征、模式的识别变得效率低下。

对VoIP应用流量进行较高效的识别是我们亟待解决的问题。如果用户通话时间较长，如何在通话开始阶段进行准确识别；如果通话时间较短，如何保证有足够的特征识别此通话流量。这些问题要求我们需要在较短时间内获取VoIP应用足够多的特征进行特征识别。

深度学习的产生为我们解决以上问题提供了思路，ZhanyiWang提出了将深度学习应用于流量识别领域的思路（The Applications of Deep Learning on Traffic Identification）。深度学习提取的特征较人为提取的特征不但具有更高的可信度，并且也大大提高了识别效率。因此，我们可以对几秒内产生的通话流量进行特征提取并高效识别。本文所提出的方法不但致力于VoIP流量的检测，同时还可以将产生VoIP流量的应用进行分类。

最后，第二节为相关工作介绍；第三节介绍使用深度学习提取应用特征；第四节介绍使用第三节中所提取的特征应用svm进行分类；第五节展示试验结果；第六节总结工作并对下一步工作进行探讨。

**2. Background and Related work**

Khan, F. I. U. A. (2008)提出了一种通用的方法进行VoIP流量检测，文章分析了整个语音流的包间到达时间（packet-inter arrival time）,包大小（packet size），包的交换率（rate of packet exchange）等特征。他们的主要目标是使用通用的特征将VoIP流量与其他流量进行区分，最终得出结论：VoIP流量在每秒内产生的数据包平均个数较其他种类的应用要多（Average packets/Sec rate is greater in VoIP as compared to other applications）；VoIP流量平均包大小要比其他应用要小（Average packet size in bytes is small in VoIP as compared to other applications）。他们的方法在检测VoIP流量的方向上起到了重要意义，但是，该方法并不能针对具体VoIP应用进行分类。

Yildirim, T., & Radcliffe, P. J. (2010, August)提出了一种致力于改善VoIP服务质量，需要较高效的进行VoIP流量识别的方法。文中提出典型的VoIP流量的数据包长度大约在60-150bytes之间。文章主要针对提高VoIP服务质量，所以需要较高效的识别出VoIP流量。但此种方法误差较大，本文也为快速并准确识别VoIP流量提供了思路。

Gomes, J. V., Inácio, P. R., Pereira, M., Freire, M. M., & Monteiro, P. P. (2013)提出了一种使用熵和编码属性来识别VoIP会话的方法。文章基于语音编码实时的对会话流进行分类，分类器并不单独的使用数据包长度，而是注重（focus）不同长度的数据包之间的关系，并在熵水平上分析其异质性（heterogeneity）。文章列出了不同应用所使用的不同编码，比较了使用恒定速率编码器和可变速率编码器的负载长度和VoIP会话前三分钟内的熵变化。文中使用了数量为500的滑动窗口来分析编码器的行为特征，设计了分类决策模型对所得行为特征进行分类。结果显示分类器对识别VoIP会话有较高的准确率，并且文中所提到的机制可以识别特定的编码器，灵敏度在70.00%-93.34%，文中提出的方法可以用于实时流量识别，但受到了未知编码器的限制。

Alshammari, R., & Zincir-Heywood, A. N. (2015)提出了一种在不包括负载的流量中提取特征集的方法，此方法基于客户端到服务器的双向流，文章使用netmate生成了数据流并对获取的双向流进行特征统计，文中列出了fiat(forward inter-arriaval time), biat(backward inter-arriaval time), fpkt(forward packet length), bpkt(backward packet length), proto, Duration等特征。通过三种有监督的方法，c5.0，Ada Boost和Genetic Programming对训练数据集进行了训练，文章结果显示skype识别率约为100%，其余VoIP应用识别率也在95%左右。文章已取得了较高的准确率，但是识别需要整个数据流的支持，不能做到实时的应用识别。

本文在以上研究基础上提出了使用深度学习的思想进行学习特征的方法。网络代理技术以及NAT技术的发展使得基于IP地址的识别受到限制，以及VOIP应用使用非特定的端口，深度学习阶段不依赖于IP地址以及端口等信息。IP地址和端口等信息在流量收集阶段以及识别阶段起主要作用。本文同时对使用不同数据包数量所得到的准确率做了研究，希望保持高准确率的同时尽最大可能减少使用的数据包，保证在通话开始阶段实时对应用进行识别。

**3. Methodology**

为了有效的在Voip通话开始阶段对应用进行识别，必须寻找一种切实可行的方法利用通话开始几秒甚至几毫秒之内的数据包对应用有效识别。而由于IP地址、端口以及加密等条件的限制，对单一数据包的识别能力是较为低下的。无法做到使用单一数据包进行识别的情况下，不得不使用多个数据包结合进行流量识别。据笔者探究，Voip应用产生的数据包个数在1秒内达数十个，这也证明我们的想法是可行的。

我们的目标是使用深度学习的技术提取多个连续数据包的特征，包括包长变化，负载类型等基本特征。本文收集了7种Voip应用的流量并对他们做了基本人为分析，有5种应用（AltCall、Jumblo、Xlite、Zoiper、UUCall）采用恒定长度的数据包，有两种应用（Skype、KcCall）采用变长数据包。其中UUCall产生的多数UDP数据包不可正确的按照RTP协议解析。KcCall产生的数据包不使用单一负载类型，虽然产生变长数据包，但数据包长度是局限于2个定制；而Skype不同，其产生随机长度的数据包。以上提到的种种行为特征，目前没有任何技术全部考虑到并进行识别，本文使用深度学习技术保证了以上全部情况的识别。为获得训练数据并准确对其贴加标签，我们使用了多种流量截取工具，WireShark、tcpdump、QPA等。我们所收集的流量来自不同时间不同地点不同网络环境。

在深度学习模型的选择上，最终将本文提出的方法部署在AlexNet网络上。

**4. Architecture**

本节我们将会介绍进行实时Voip流量识别的基本结构，并对结构中的2个阶段做基本介绍。本文所实现的结构可以针对加密的voip应用自动生成特征用以实时的识别voip应用流量,包括训练和识别两个阶段。

i. Train

训练阶段致力于使用上文提到的7种Voip应用流量训练深度学习模型，找到最合适的函数保证最高准确率识别。训练阶段包括数据预处理、训练模型、评估模型4个过程。本文采取的有监督的学习方法要求数据预处理需要对获取的流量数据进行生成带标签的数据；数据预处理还需要将数据包格式转化为矩阵的格式用以输入深度学习模型，其中要对数据包进行剔除网络层和传输层头部的操作；最后数据预处理还要进行矩阵归一化操作。训练模型求归一化后的矩阵和对应的标签作为输入，进行不断最小化损失函数后得到识别模型。识别模型即我们用来voip应用识别的最终模型。最后评估模型是用来评估最终检测模型质量的过程，通过损失函数和识别准确率进行评估，同样使用归一化矩阵和对应标签作为输入。

ii. Identification

识别阶段是voip流量识别的最终环节，通过训练的的模型对实时监测到的voip流量进行识别。其包括rtp数据包检测、数据包处理、识别3个过程。Rtp数据包检测过程要求使用ip地址以及udp端口确定voip通话连接以及源ip地址和目的ip地址进行监测。确定一个voip连接之后，进行数据包处理，需要剔除网络层和传输层头部并归一化处理。将处理后的矩阵输入识别模型后即可以得到识别结果。

**5. Traffic Collection and Processing**

为了训练voip识别模型，我们需要收集大量加密voip应用流量。为了获取多种情况下的流量，我们在校园网络内部以及外部部署了voip软件并在不同时间段进行流量捕获工作。

基于有监督的学习，为了尽可能准确的对流量进行贴标签的工作，我们在windows系统下使用基于进程的抓包工具qpa；在linux系统下使用tcpdump基于端口进行抓包,待查看进程占用端口后，指定tcpdump按端口进行流量捕获。额外分析工作使用广为人知的wireshark网络封包分析工具进行。我们截取了7中加密voip应用的流量数据如表1所示，因为基于进程和端口的抓包工作，我们很容易对其进行贴标签的工作。

|  |  |  |
| --- | --- | --- |
| **VoIP application** | **Platform** | **Size** |
| **Skype** | **windows and linux** | **1454.4M** |
| **Uucall** | **windows** | **854.7M** |
| **Kccall** | **windows** | **1064.4M** |
| **Altcall** | **windows** | **897.6M** |
| **Jumblo** | **windows and linux** | **1352.4M** |
| **zoiper** | **windows and linux** | **1709.1M** |
| **xlite** | **windows and linux** | **2082.6M** |

在输入深度学习模型进行训练之前，我们需要对流量进行预处理的工作：1）对流量数据包除去网络层和传输层头部；2）将多个连续数据包按照ascii码转化为矩阵；3）进行归一化矩阵操作。

本节将会介绍研究中所涉及的数据以及方法等内容。研究中主要针对国内常用的5种VoIP应用进行分类，深度学习过程采用Googlenet网络进行训练及评估。

3.1 Labeling Dataset

在收集VoIP应用流量的过程中，本文研究人员做了大量工作。生成带标签的大规模的数据在研究过程中一直是最大的难题。本文在学校网络环境中使用tcpdump软件按照五元组（源ip、目的ip、源端口、目的端口、应用协议）收集了国内常用的五种VoIP应用（skype、uucall、kccall、altcall、jumblo）的数据包，获得了如下pcap文件：

|  |  |  |
| --- | --- | --- |
| **VoIP application** | **Pcap file** | **Size** |
| **Skype** | **Skype1(1-12).pcap** | **210.9M** |
| **Uucall** |  |  |
| **Kccall** |  |  |
| **Altcall** |  |  |
| **Jumblo** | **jumblo(1-59).pcap** | **112.7M** |

本文使用Googlenet网络对带有标签的数据使用有监督的学习方法进行训练，我们对捕获的流量数据进行预处理操作，将带标签的数据按照图片的格式保存，如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VoIP app** | 56 | 112 | 224 | Lable |
| **Skype** | (1-44800).png | (1-22400).png | (1-11200).png | 0 |
| **Uucall** |  |  |  | 1 |
| **Kccall** |  |  |  | 2 |
| **Altcall** |  |  |  | 3 |
| **Jumblo** |  |  |  | 4 |

3.2 Dataset processing

在3.1节我们介绍了所使用的数据集以及对数据集做的二次格式的保存方式。本节我们将会介绍对收集的pcap文件的具体处理方式。我们知道，pcap文件是以16进制的格式保存原始数据包的格式。基本单位为256个十六进制的ascii码（00-FF），而灰度图（Gray Scale Image）按照灰度也分为256阶，因此本文提出将数据流量保存为灰度图以便下一步处理。因为本文所提出了实时分类的方法不依赖于IP地址以及端口等信息，所以在数据集处理阶段我们只保存整个数据包第42个字节之后的数据。

根据以上规则，我们按照数据包个数进行处理，分别为56，112，224等。主要目的是为了寻找可以达到最高识别率所使用的最小数据包数。

3.3 deep learning

深度学习方向有很多优秀的网络结构。诸如Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012)提出的AlexNet；Simonyan, K., & Zisserman, A. (2014)提出的VGG网络；Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015)提出的GoogleNet；He, K., Zhang, X., Ren, S., & Sun, J. (2016)提出的ResNet网络。

googlenet网络是在ILSVRC2014中提交的版本，是一个22层的高质量模型。Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015)等人认为获得高质量模型的最佳方案就是增加模型的深度和宽度。但是在增加深度和宽度的同时会带来很多问题。文章采用Inception架构解决了增加宽度的同时所带来的参数大幅增加的问题，Inception架构是一种加入1\*1卷积核进行有效降维的结构。另外，文章在网络的不同深度处加入了两个loss来避免深度增加所造成的梯度回传消失的现象。

本文采用随机梯度算法(Stochastic Gradient Descent)来最小化训练样本的损失函数

**6. Learning using AlexNet**

**5. Performance evaluation**

**6. Conclusion**

**7. Acknowledgments**

本文主要对加密的voip应用进行分类操作，学习的过程采用有监督的学习方法。传统的机器学习方法需要人为的去创建特征集，此类方法不但效率低下，并且其准确率没有保障，如采用特征平均包长度来作为特征进行分类，当碰到两个应用的udp报文长度相等时（如本文uucall和altcall）准确率会大大降低。前面提到，本文采用深度学习的思想去获取流量特征，减少人为工作的同时也提升了分类准确率。

**8. Reference**

[1] Wang, Z. (2015). The applications of deep learning on traffic identification. BlackHat USA.

[2] Khan, F. I. U. A. (2008). A generic technique for voice over internet protocol (voip) traffic detection. IJCSNS, 8(2), 52.

[3] Alshammari, R., & Zincir-Heywood, A. N. (2015). Identification of VoIP encrypted traffic using a machine learning approach. Journal of King Saud University-Computer and Information Sciences, 27(1), 77-92.

[4] Yildirim, T., & Radcliffe, P. J. (2010, August). VoIP traffic classification in IPSec tunnels. In Electronics and Information Engineering (ICEIE), 2010 International Conference On (Vol. 1, pp. V1-151). IEEE.

[5] Gomes, J. V., Inácio, P. R., Pereira, M., Freire, M. M., & Monteiro, P. P. (2013). Identification of peer-to-peer voip sessions using entropy and codec properties. IEEE Transactions on Parallel and Distributed Systems, 24(10), 2004-2014.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).