Content Based SPAM Detection and Filtration Prototype for VoIP

Justin Nafe
College of Engineering
Computer Science and Engineering
University of North Texas
jjn0032@unt.edu

Ben Raybon
College of Engineering
Computer Science and Engineering
University of North Texas
bbr0006@unt.edu

Hieu Nguyen
College of Engineering
Computer Science and Engineering
University of North Texas
hieunguyen95@hotmail.com

Abstract

Group Seven addresses the susceptibility of abusive marketing (SPAM) and scamming over a Telephony system. Group Seven created a SPAM filtering prototype for Voice over IP (VoIP).

VoIP and Session Initiated Protocol (SIP) creates a new vulnerability to telecommunications, because VoIP, a nearly free avenue for communication, appeals to business owners and marketers who may have the intention to use Dialer applications to make calls in bulk. This makes the VoIP protocols susceptible to SPAM.

The design is similar to content packet filtering, in that the content of incoming packets are analyzed, and the design is similar to an email SPAM filter, in that voicemails are classified as SPAM or not SPAM based on the history of what the program knows as SPAM (Fisk 2001, Meyer 2004). The prototype uses Bayesian probability for this learning and to calculate the likelihood that the voicemail is SPAM..

Introduction

VoIP started out in 1996 in the practical sense with a shaky start, but picked up in 1998 (History of VoIP). VoIP provides users with a free or inexpensive avenue to communicate through the internet as if making a telephone call. This relatively new and inexpensive technology opens up a new avenue for spammers. The only regulation attempting to prevent the abuse of phone calls is the CAN-SPAM Act of 2003 and the National Do-Not-Call Registry. The exceptions are from campaign and charity organizations (Cell-phone spam: How to curb it). Possibly due to the newness of this technology, the abuse of VoIP is not widespread, but given that VoIP is inexpensive and an avenue to market, a Dialer, numerous DIDs, and a marketing recording, such as an advertisement for Viagra is all that is needed to abuse VoIP. For example, email is an inexpensive service, an ideal avenue for marketers. Once bulk email tools were developed, SPAM became the abuse of email marketing. Our program attempts to capture, tag, and learn voicemail SPAM (voicemail recordings of a recording marketing a service that the receiver did not want).

System Setup

Asterisk is an Open Source Telephony system that uses the internet to transmit voice (http://www.asterisk.org/about). We installed Fonalityôs Asterisk based Trixbox. (a free PBX, but Fonality charges for support if the user chooses). We installed Trixbox on an obsolete computer in a

remote location. Asterisk reformats the entire hard drive, so we had to install Trixbox on a standalone computer (Trixbox does not give you any chance to partition your hard drive, so if you try this at home, just be aware of this feature). After installation, an update was necessary to enable our softphones to work with the system. We connect and make and receive internal calls with the CounterPathôs X-Lite softphone. To be able to make calls outside of our LAN or to a regular phone line, we would need a Direct Inward Dialing (DID) number, a SIP channel (provided by a third party or by setting up another server as an Internet Gateway), and a domain name for my dynamic IP address. Though these features add interesting functionality, it is unnecessary for our project. This setup also keeps the environment simple and negates any other complications that the other features may cause. X-Lite is freeware for personal use and can be downloaded at http://www.counterpath.net/X-Lite-Download.html.

System Configuration:

To be able to connect to the server for development purposes, we installed ssh, and for configuration purposes, we mostly used the website provided with Trixbox for setting up extensions. The following ports needed to be opened on the network where the server is stored in order for external phones to communicate:

- SIP: 5060 TCP/UDP
- RTP 10000 to 20000 UDP
- MGCP 5036 UDP

The following programs needed to be installed for development of our program. To compile our program on the VoIP server, we needed compilers and interpreters, particularly Perl (v5.8.8 built for i386-linux-thread-multi) and a C compiler (gcc 4.1.2). Tcpdump (version 3.9.4) and Libpcap (version 0.9.4) needed to be installed to capture the packets. We ran into many problems that

OpenVPN (for connecting from the wireless network at the research park)

PKI ï setup for a public key and private key with a master Certificate Authority.

Generate a master CA certificate/key, a server certificate/key, and certificates/keys for 3 separate clients (http://openvpn.net/index.php/documentation/howto.html#install).

Roy from DataComm made a special exception to allow the SIP protocol through the wireless network at the research park.

Presentation

Preferably, two X-Lite phones will be used for the demonstration, one registered to one extension and the other registered to a different extension. One problem with this is that outside of the LAN that the server is on, voice does not transmit. Speaking with a DataComm Support Technician (DataComm handles our wireless connection at UNT, and the blocking of ports) and knowing about UDP and the lack of handshaking, our voice packets get lost since I am behind a NAT and the server is behind a router. In other words, the packets donôt know where to find me. . To eliminate the chance of any ports being blocked and interfering with our connection to the server, the support technician is allocating a Public IP address to my PC. networkThe phone looks like the following screenshot. Establish a call between the two with no interference to hear how clear it is, listening for jitter (dropped packets), etc.

Establish a call running the program (open source or tcpdump) to hear how clear it is now and to capture normal packets.

Establish a call when the system is under attack, capturing the packets and making a note on how it sounds.

Run the program to detect abnormalities or a ñbadò call.

Go through the same process as above except try to predict the abnormalities, or catch the abnormalities when they happen and either alert the user or deal with the problem accordingly.

Program

The Context Based Intrusion Detection system consists of two parts. One part enables the user to tag a voicemail as SPAM and puts the SPAM voicemail in the SPAM folder of the userôs extension. We call this the supervised learning aspect of the program. The other part of the program (we will call this part the Filter), scans the SPAM folder of each extension for any new files marked as spam. If the program detects a new file, then the program filters out the stop words (words, such as ffff and 0000, that occur in every voicemail and that occur many times in each file) and puts the remaining words into a hash table, tracking the occurrences. The testing section of the prototype uses this information for the calculation of the Bayesian classifier. The testing section classifies an email as SPAM or not SPAM, using the Bayesian algorithm, which takes into account the history of occurrences of the feature words gathered by the training section. The testing section constantly scans each extensionôs Inbox for new voicemails, and when one is found, the testing section processes it using Bayesian mathematics to classify the probability that the voicemail is SPAM. If the program determines that the voicemail is SPAM, then the program moves the voicemail to the SPAM folder for the training section to learn from. This is the learning of our system. Determining whether or not the voicemail is spam is based on probability. If the voicemail contains frequent patterns that are also found in the SPAM files (stored in the hash table), then the likelihood of the voicemail being SPAM increases. One of the drawbacks of this method is that the program needs to learn in order to accurately classify voicemails. In other words, the more examples it has of ñwhat is a SPAM exampleò the better it will be in classifying future voicemails.

Related Work

Our program mostly relates to the filtering of SPAM, except the media is voicemail instead of email. For more of an Intrusion Detection System for the SIP handshaking, refer to the report by Y. Ding (Ding).

References

CAN-SPAM Act 15 U.S.C. 7701-7713 (2003)

Cell-phone spam: How to curb it. http://blogs.consumerreports.org/electronics/2008/03/cell-phone-spam.html March 12, 2008

GFi White Pater. Why Bayesian filtering is the most effective anti-spam technology. GFI Software, 2007.

History of VoIP. University Texas Dallas. http://www.utdallas.edu/~bjackson/history.html 2004.

http://www.asterisk.org/about.

http://www.trixbox.org/ Trixbox CE. Fonality 2008

J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. RFC 3261 SIP: Session Initiation Protocol. *In Network Working Group, The Internet Society June 2002*

LZO (Library files used by OpenVPN) Copyright (C) Markus F.X.J. Oberhumer. http://www.oberhumer.com/opensource/lzo/

M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. RFC 2543 SIP: Session Initiation Protocol. *In Network Working Group, The Internet Society June 1999*

Mike Fisk, George Varghese. Fast Content-Based Packet Handling for Intrusion

Detection. *In UCSD Technical Report CS2001-0670, May 2001*

National Do-Not-Call Registry. FCC. http://www.fcc.gov/cgb/donotcall/ last reviewed/updated on 05/22/08

T.A. Meyer and B. Whateley. Spambayes: Effective open-source, bayesian based, email classification sytem. In *First Conference on Email and Anti-Spam,* July 2004

The Softphone Application used came from: (http://www.counterpath.com/home.html) Copyright © 2003 - 2008 CounterPath Corporation.

The VPN Connection used to bypass the wireless firewall at work is OpenVPN and came from http://openvpn.net/

Yanlan Ding, Guiping Su. Intrusion detection system for signal based SIP attacks through timed HCPN. In *Second International Conference on Availability, Reliability and Security (ARES'07) 2007.*