Analysis of SMS Spamming Solutions

Wynne. Lui, Vincy Tang, and Jue Ni

***Abstract*— Short Message Service (SMS) is a protocol that allows mobile phone users to send short text messages to servers, other users and telecommunication service providers. SMS spamming can create annoyance to end users, disrupt availability of SMS services, and create delay or denial of services. In this report, we analyze different ways malicious parties can execute SMS spamming attacks at both the user and the server level. We study the existing solutions and propose a more economical and platform-independent solution using open-sourced software package and scripting language. A prototype Perl script is developed to prove our concept. With advancement in technology and increasing number of GPS-enabled phones, there could be further improvements to our current solution.**

***Index Terms*—SMS, spamming, gateway**

I. Introduction

SHORT Message Service (SMS) allows mobile phone users, telecommunication service providers and some small businesses to send short text messages of the limits of 160 characters. The communication is achieved by using one or all of these devices and media: mobile devices, computers connected to the Internet, SMS gateway. SMS gateway is an important medium that allows a computer or sever to send and receive SMS messages to or from a mobile device.

In this project, our main objectives are to analyze different ways of how malicious parties could attack any device or medium listed above using SMS spamming, propose an economical solution that can detect SMS spamming at the gateway level and design a prototype script to prove our concept.

SMS spamming can be a very serious issue with the increasing number of mobile phone users and SMS telemarketing adopters. More and more companies have become aware of advertising through SMS, which is much cheaper than traditional advertising methods. [10] SMS can be also used by malicious parties to attack other SMS gateways. These gateways could be used by SMS service based companies. This is a serious threat since it is quite simple for a user with basic computer knowledge to set up a personal SMS gateway and set up an automatic script to attack other gateways.

In some countries, like Australia, China and Korea, the local government has already become aware of unauthorized SMS telemarketing and spamming. [2] Although the same problem has not been elevated to the same level in Canada yet, it could become serious with the forecasted increasing number of text message users. [10]

In our report, we will address different spamming attacks by discussing the level of severity, current solution(s) and possible improvements. We will focus on one attack that is least addressed with current work and design an economical solution that can be easily integrated into any existing platform.

II. User Level Attack

A. HTTP Spam Attack and Solution

In HTTP and Email spamming, the spamming is generally done on the user level. Http can be used to send SMS with a web application, using the internet. These include portal set up by the service provider to send SMS through their website or through instant messaging programs. To send a text message, only a cellular phone number is required.

To launch a spam attack, only access to the web application and a list of cellular phone numbers is needed. Attacks can be made at any location with access to the internet, and since a login is not necessary, everyone has access to these web applications. To obtain a list of phone number to attack, a user can easily generate lists of random valid phone numbers. It is not necessary to avoid attacking non cellular phone numbers since those messages will just be ignored within the network. When a text message is sent from a web application, the sender's number is within a list of trusted party. This is because an HTTP gateway must be registered with HTTP service carriers . This type of spamming is a problem because the end user must pay, incoming text messaging and roaming, when receiving each text message sent via HTTP.

The solutions necessary to prevent SMS attacks via HTTP are currently implemented by the application providers. When using the application from the service provider, the submission is immune to scripts since they use a verification process that cannot be done with a script. This verification is done by providing a picture of letters then requiring the user to type in the letters. This process is being implemented when sending messages through Rogers Communications' website. When an attacker cannot use a script to send messages, it makes it infeasible to launch an attack since it would be inefficient and would take up too much time. In the instant messaging applications where it is possible to send text messages, the end user must register and give permission that messages sent through the application be routed to their mobile devices. This removes liability for the application providers since the user accepts the risk when granting this permission. These steps are taken by the application provider because they must register their gateway. They must ensure

their gateway cannot be easily exploited to retain their registration.

B. Email Spam Attack and Solution

The Email to text messaging service is provided by the service provider. Each cellular phone number has an email address where the mail messages received are sent to the mobile device as a text message. The format of email address is strictly defined by the service provider and only requires the cellular phone number. Similar to a HTTP based attack, the attacker can easily generate a valid phone number to spam. Also, the end user may get charged for receiving email sent text messages. Since these messages are sent via email, current email spam filters are advanced enough to filter out these spam attacker's messages. Some service providers require a subscription to receive these email messages on their mobile devices.

C. SMS Telemarketing Attack and Solution

SMS telemarketing is also another channel for SMS spamming. As the number of mobile users increases, the general population would have more access to a mobile device than to a television. This will make it feasible for businesses to promote their products via SMS messages.

The configuration required for business to send SMS messages is very minimal. All that is required is a computer connected to a mobile device with prepaid card and a list of valid phone numbers. The mobile device acts as a modem to connect with the cellular network. These businesses then can use open source applications, such as Gnokii [9] , that allows sending bulk messages. A prepaid card is approximately $150 and sometimes service providers provide first month unlimited text message services to these customers with prepaid services. In comparison to traditional methods of advertising, this is an extremely more economical method to promote their products. Once the computer and phone is configured, the advertiser can easily run a script to send bulk messages to phone numbers that was generated using a simple script. To generate these numbers the user can specify valid mobile phone prefixes, such as:

Area Code	Phone prefix:
778	XXX-XXXX
604	218-XXXX
604	808-XXXX
604	671-XXXX
604	710-XXXX
604	889-XXXX

Table 1 - List of Know Cellular Phone Number Prefixes

As long as the numbers are a valid phone number, these advertising messages will be delivered to the owner of this phone number. If the number is invalid, they message will be dropped in the SMS gateway. This can be a potential problem since it uses a script to send a large amount of messages through one gateway. This can cause congestion on the network and create a denial or delay of service to other users.

Currently, the solutions available are targeted towards selected device models or are too expensive to be afforded by the common consumer. For example, Fortinet developed a software solution that supports only a selected few Personal Digital Assistant (PDA) or smart phones. [8] This is does not protect the majority of mobile users with other kind of devices. There are also some deterring government policies to protect against illegal telemarketing. [4] Cisco has also included a spam filter feature on some of their routers. This can be used by the service providers to protect their customers against illegal spamming. [5]

III. Server Level Attack

A. SMS Service Spam Attack

Business that provides their services via SMS could be the target of SMS attacks. One example of such service is The Weather Network, which provides local weather information to users through a two-way text message service. Users send requesting text messages to a server, which processes the information and replies with another text message.

Business needs to set up its own SMS gateway server that handles incoming and outgoing text messages. Delay or denial of service can happen when an attacker tries to send malicious text messages to the server to overwhelm the server. This attack can be easily achieved by setting up the attacker's own gateway using a computer, a mobile phone and an open-sourced package like Gnokii. [9] An automated script can then be written to keep sending messages to one or multiple cell phone numbers in an infinite loop. Below is a sample code that illustrates how the attack works. It is written in Perl using the Gnokii-smsd daemon and mysql database library.

```
while (true)
{
 ## prepare database
 $db_handle =
  DBI >connect("dbi:mysql:database=PriceDB;host=localhost;
  user=apsc486;password=price") or die "Couldn't connect
  to database: $DBI::errstr\n";

 print "$numbers $spam_text, $phone \n";
 #$sql = "select * from outbox";
 $sql = "INSERT INTO outbox (number,text,phone)
  values ('$number','$spam_text','$phone')";
 $statement = $db_handle->prepare_cached($sql)
  or die "Couldn't prepare query '$sql':$DBI::errstr\n";
 $statement->execute()
  or die "Couldn't execute query '$sql': $DBI::errstr\n";

 ## Output message
 print "-------------------------------------------------\n";
 print "-    sending messge $index to number $numbers   -\n";
 print "-------------------------------------------------\n";

 ## put 10 seconds difference between each message to per number
 print "-------------------------------------------------\n";
 print "-         waiting for 10 seconds               -\n";
 print "-------------------------------------------------\n";
 sleep 10;
}
```

B. Existing Solutions

Some router companies have products with an anti-spamming filter feature. For example, Cisco has released the

Cisco® SMS Spam and Fraud Prevention Solution that can be used to protect the servers. Unfortunately, some small or start up companies cannot afford to use this solution.

C. *Proposed Solutions*

We are proposing to develop an economical and platform-independent solution that could be easily adopted by regular developers work in small businesses that provide premium text message information. This solution can be achieved developing a Perl script utilizing a software tool called Gnokii.

Gnokii is an open-sourced SMS gateway package is used by a lot of developers in the open source community. It is well updated and supported, and works with different platforms like Linux, Windows, MacOS and Unix. The Gnokii-smsd daemon enables a developer to send messages using command lines, and links all messages to mysql database. All incoming and outgoing text messages to the gateway server are stored in the inbox and outbox tables respectively.

Our solution adds a filter to the gateway by auditing and analyzing all incoming messages to determine whether they are malicious. The timestamp and the source phone number of each incoming message is logged for further comparison. In the analysis process, the following two actions are performed.

1) Compare the source phone number to list of numbers in the black list.
2) Compute the time difference between the current and the previous messages if they were sent from the same source phone number.

The message is categorized as a spam message if it meets any of the following set of criteria.

a) The source phone number is on the black list
b) The time difference is less than 30 seconds

If the message met b) but were not in the black list, the source number would be added into the list. The time different limit is a pre-set constant that can be changed by the user according to the severity of the spam attack. Figure 1 below is a flowchart of the filter script.

Included in the appendix is a sample code that illustrates how the filter can be implemented in Perl using Gnokii-smsd and mysql commands.

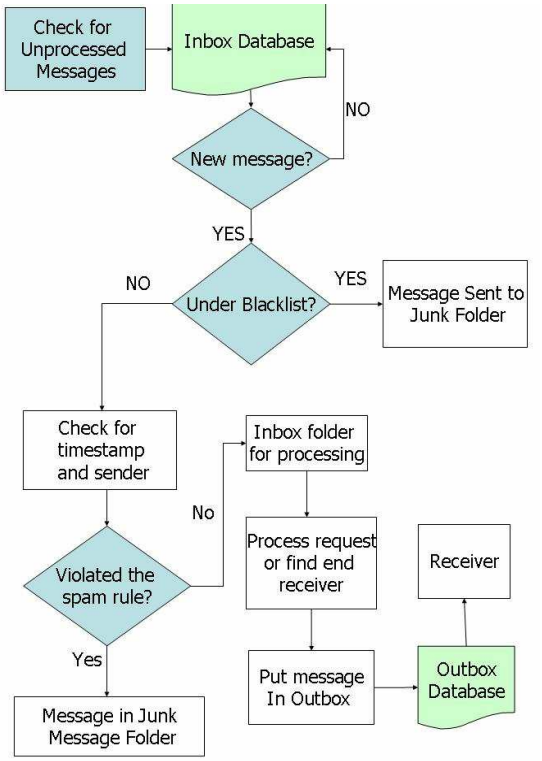

Figure 1 - Filter Algorithm Flowchart

IV. Conclusion

Compared to existing solutions, our proposed solution can be implemented easily at the server level with very low cost and on different platforms. The script can be easily integrated into the gateway without any user involvement. Because the script is developed using an open-sourced API tool, the cost of development is almost zero. The drawback is that there is no guaranteed technical support or updates, so it is more applicable to small businesses that are willing to endure instability for lower costs.

Overall, this prototype can be contributed to the open source community for developers that use Gnokii in their own applications. The script can be run on different platforms without additional set-up. It can also be easily translated into other scripting languages and customized according to the specific application.

The current solution only checks the timestamp and the source phone number of each message, because these two fields are the most important fields in the current SMS gateway system. The solution will become more complicated if GPS-enabled mobile devices were more widely used in Canada. Currently, location-based SMS spamming has already become a problem in Asia where most mobile devices are GPS-enabled and location information can be easily accessed by a third party. [10] In the future, all phones in Canada are going to be equipped with GPS chips. With a user's consent, it will be possible to provide location based text message service. Sensitive information like mobile user's location will be exposed to attackers and spammers. To prevent similar problems, a more complicated filtering system

will need to be implemented in addition to more stringent government regulation on SMS spamming. In addition, the location information might need to be encrypted to protect the end users.

APPENDIX

Below is the sample code of how a filter can be implemented in Perl using Gnokii-smsd and mysql commands

```
# Prepare output text file
open(sms_in, ">> sms_in_copy.txt")|| die ("Could not open file!");
open(bl_log, ">> black_list.log") || die ("Could not open file!");
## Processing new messages
while ($row_ref = $statement->fetchrow_hashref()){
 ## Get message info
 $number = $row_ref->{number};
 $time = $row_ref->{smsdate};
 $time =~ s/\n|\r|\-|\s|\://sg;
 $text_info = $row_ref->{text};
 $text_info =~ s/\n|\r//sg;
 $id = $row_ref->{id};
 ## check number with black list numbers by searching through the list (
 if (isBlackList($number) == 1){
  $new_info = "$id<$number<$time<$text_info\n";
  print bl_log ($new_info);
  print "----------------------------------------\n";
  print "-       Got spam messages from     -\n";
  print "-       $number on black list      -\n";
  print "----------------------------------------\n";
 }
 else{
  ## compare timestamp and phone number
  if(((($time-    $prev_time)<=$TIMEOUT)&&($prev_number==$number)
){
   ## if the program goes here, then the message and the previous one
should
   #be put into the blacklist open file to be written
   open(black_list, ">>black_list.txt")||die("Could not open black list
file!");
   if ($prev_logged == 0){

$prev_info="$prev_id<$prev_number<$prev_time<$prev_text_info\n";
    print bl_log ($prev_info);
    $prev_logged = 1;
    print "----------------------------------------\n";
    print "-       Got spam messages from        -\n";
    print "-       $prev_number at time       -\n";
    print "-          $prev_time            -\n";
    print "-          $time               -\n";
    print "----------------------------------------\n";
   }
   $new_info = "$id<$number<$time<$text_info\n";
   print bl_log ($new_info);
   print black_list ("$number\n");
   print "----------------------------------------\n";
   print "-       Got spam messages from        -\n";
   print "-       $prev_number at time       -\n";
   print "-          $prev_time            -\n";
   print "-          $time               -\n";
   print "----------------------------------------\n";
   ## close log files
   close(black_list);
   }else{
   ## log into text file in the format of "number<product and other text "
   if ($prev_logged == 0)
   {
    ## check for postal code in the format v5t1e5/112456
    # or v5t/345 or v5t 1e5/112 356
    if (($text_info =~ /(.+)(\s\w\d\w\d\w\d)+$/)
      || ($text_info =~ /(.+)(\s\w\d\w)+$/)
      || ($text_info =~ /(.+)(\s\w\d\w\s\d\w\d)+$/)){
     $text_info =  "$1<$2\n";
    }else{
     $text_info = "$text_info<NOPC\n";
    }
    ## log into text file in the format of "number<product and other
    #text<postal code/NOPC if no postal code"
    $new_info = "$row_ref->{number}<$text_info";
    print sms_in ($new_info);
    print "----------------------------------------\n";
    print "-       Got message from $number      -\n";
    print "-        Message processed            -\n";
    print "----------------------------------------\n";
   }else
   {
    $prev_logged = 0;
   }
  }
 }
 $prev_time = $time;
 $prev_number = $number;
 $prev_text_info = $text_info;
 $prev_id = $id;
 }
close(sms_in);
close(bl_log);
```

REFERENCES

[1] A. Dickinger, et al "An Investigation and Conceptual Model of SMS Marketing", 37th Hawaii International Conference on System Sciences, 2004.

[2] Australian Government, "Spam Regulations 2004", Apr 2007. Available: http://www.comlaw.gov.au/ComLaw/Legislation/LegislativeInstrumentCompilation1.nsf/0/ED5984D617843AB1CA256F7100590CEB/$file/2004No56.pdf.

[3] B. Deakins, U. Grandcolas, and R. Rettie, " Text Message Advertising: Dramatic Effect on Purchase Intentions", Apr 2007. Available: http://www.uniqueadvertisingstrategies.com/Text%20Message%20Advertising%20Dramatic%20Effect%20on%20Purchase.pdf

[4] BBC News, "Mobile spam on the rise", Apr 2007. Available: http://news.bbc.co.uk/1/hi/sci/tech/2116070.stm.

[5] Cisco Systems Inc., " SMS Spam and Fraud Prevention", Apr 2007. Available: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns278/c654/cdccont_0900aecd80250cb6.pdf

[6] Free Patents Online, "Method for blocking spam messages in a mobile communication terminal", Apr 2007. Available: http://www.freepatentsonline.com/20050020289.html.

[7] Free Patents Online, "Spam filtering for mobile communication devices", Apr 2007. Available: http://www.freepatentsonline.com/20060041622.html

[8] Fortinet, "FortiClient Datasheet", Apr 2007. Available: http://www.fortinet.com/doc/vpn/FortiClientVPN.pdf.

[9] Gnokii, "Documentation", Apr 2007. Avaliable: http://www.gnokii.org/docs.shtml

[10] X. Heng, et al, "Foundations of SMS Commerce Success: Lessons from SMS Messaging and Co-opetition", Apr 2007. Avaliable: http://csdl2.computer.org/comp/proceedings/hicss/2003/1874/03/187430090b.pdf