Global Information Assurance Certification Paper

SANS Parliament Hill 2000

GIAC Advanced Incident Handling and Hacker Exploits Curriculum Practical Assignment

Option 2 – Document an exploit, vulnerability or malicious program

Adrien de Beaupré

September 2000

Introduction:

What I would like to describe are two of the (many) security issues currently facing security and systems administration personnel. They both involve the users at home or from within a network exposing themselves to risk. The first issue is the spread of 'peer to peer' style services among Windows users on the Internet, virtually pandemic. Examples include Napster, Gnutella and ICQ, among others. These are all designed to allow users to either communicate directly or share data with each other. They are not designed with security in mind and are instead adept at avoiding the servers and systems that administrators may have put in place as protection. The second is the exploitable buffer overflow vulnerabilities that are often found in these programs. So installing ICQ will actually open up further vulnerabilities above and beyond those that exist already when using Windows based operating systems on the Internet. ICQ is not usually perceived as a security problem either by users or administration and is quite often ignored. In some cases I have seen ICQ used within an organization as an official communication media. ICQ communications will quite often bypass firewalls, Intrusion Detection Systems and other security measures. The protective measures are facing outwards and do not prevent this risk because the threat quite often begins from within, not the wild and wooly Internet. The users see these tools as fun, convenient or useful and do not tend to see or believe the potential for damage to their own computer and those around it. Once exploited victim computers can be used as starting points for other types of attacks. A clear example would be exploiting ICQ to introduce a series of back door Trojans and leapfrog attack from there. Once that door is open the attacker has free reign to your systems and networks. They can install DDOS tools, sniffers, password crackers, access sensitive data and so on.

I have chosen ICQ for the purposes of documenting one of the vulnerabilities and exploits possible with instant messaging. It is susceptible to a fairly large number of other attacks as well. This specific exploit involves sending someone on ICQ a URL link within a message, causing a buffer overflow which can result in malicious code executing on the victim computer.

Exploit Details:

Name(s): ICQ URL Remote Exploitable Buffer Overflow

Bugtraq: 20000111 ICQ Buffer Overflow Exploit

ISS X-Force: icq-url-bo (3889) ICQ Incoming URL Buffer Overflow

CVE: CAN-2000-0046 (under review)

Variants: None (See additional ICQ vulnerabilities and exploits)

Operating Systems Impacted: Any version running ICQ
Mirabilis ICQ 0.99b v.3.19 and 1.1.1.1
Win 3.X, Win 9X, NT 4.0, Win2K.

Open source variants of ICQ, and the Java based client running under other operating systems do not appear to be vulnerable to this particular exploit

Protocols / Services: TCP/IP, multiple versions of the ICQ Protocol

Brief Description: ICQ can be vulnerable to an exploitable buffer overflow condition using a long URL with some malicious assembly code appended on the end. The URL is transmitted in a message and the code may execute if the user clicks on the link.

Exploit type: Denial of Service, Remote execution of code

Tools: Any version of Microsoft Windows running ICQ client.

Protocol Description:

ICQ is used as an instant messaging system over TCP/IP. The client software is downloaded and installable for most operating systems. It is a very popular chat style program, currently freely available on the Internet from Mirabilis (Acquired by AOL 1998). ICQ is an 'all-in-one' software package. The main purpose is to be notified if one of your friends is online, then allow you to send and receive messages. It also has a built in IRC style chat program, email client and web server. To begin with the client will register with the central ICQ server. Once the ICQ program detects an Internet connection it will register the UIN (Unique ICQ Number, or Unique Internet Number) of the user with a directed UDP packet to port 4000. This registration packet is acknowledged by the server to the client over UDP. Once a user is online others can communicate and send messages. The clients will use TCP to establish a session. The TCP ports used are usually between 1024 and 2000. The ICQ protocols are proprietary and have not been released for verification and testing, most of the ICQ protocol specifications available are speculation base on packet analysis and reverse engineering . The only security appears to be at the client, it will verify the password locally, and then transmit it to the server, but not for authentication. Earlier versions of the protocol, from 1 to 3 did not employ any forms of verification or security at all. Versions 4 and 5 do use a form of encryption, random session ID numbers and communication sequence numbers to introduce a limited form of security to ICQ.

Description of variants:

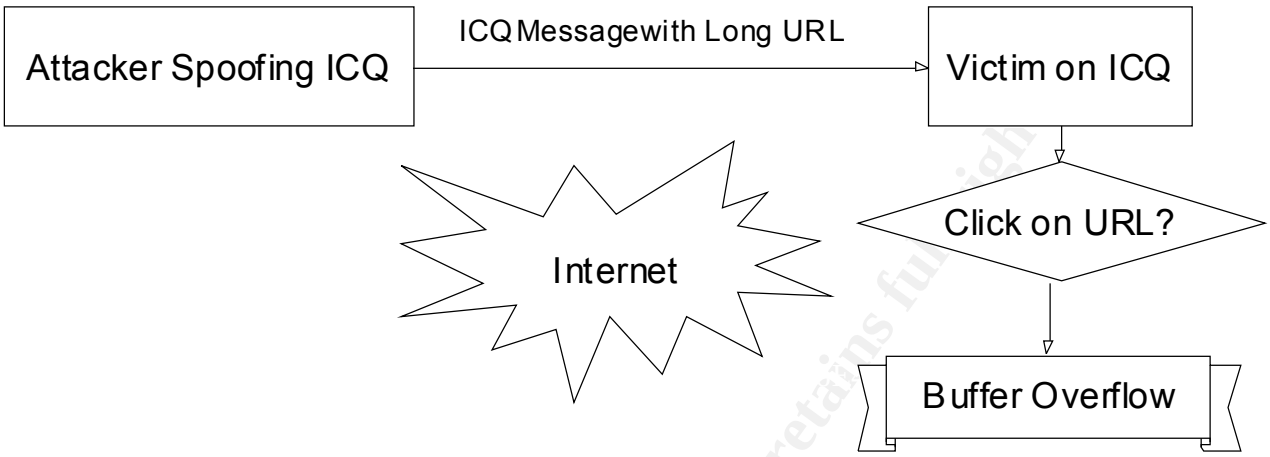
There do not appear to be any specific clones or copycats of this particular exploit in the wild. However, there are any number of other vulnerabilities within the ICQ protocol and the client software to choose from. At the end of this document there is a brief description of some of these. The main vulnerability I am discussing is exploiting a buffer overflow condition due to the lack of bounds checking. The announcing posts on BugTraq also mention that the ICQ client software does not seem to do any data checking at all, so there are likely to be a large number of other exploitable areas.

How the exploit works:

Briefly, the attacker will copy a long URL and then paste it into an ICQ message sent to a victim. Sending the message will not cause the exploit. The user receiving the message has to view the contents of the message and then click on the URL. Therefore an action is required by the victim for this to work. At that point ICQ will parse the URL and attempt to pass it to the browser. ICQ does not check on the length of the URL, the buffer overflows and characters at the end can be executed. If the code is assembly language it can do pretty much anything. When tested on Windows 98, Windows NT 4.0 (SP6a) and Windows 2000 Professional in my lab the exploit crashed ICQ and occasionally their TCP/IP stacks, requiring reboots. Usually the Windows 98 computers simply froze up. They were all running ICQ version 99b 1.1.1. Using the sample code, more of a proof of concept, which does not contain any malicious code this performs a simple denial of service. ICQ will not allow you to type that many characters into a message. It will allow as many characters as you want to be pasted into a message if they start with 'HTTP://' . So there is no limit to the length of a URL sent in a message to our unsuspecting victim, you would of course start it out with "Check out this really cool site and click on this!". The length of the string is longer than the buffer it is sent into can handle. The overflow over-writes the EIP, or return address, which will now point back to where the malicious exploit code is in the stack. Known as 'smashing the stack' this is a brief description of how a buffer overflow exploit would work.

- Send a string of data longer than the buffer receiving it can handle
- No bounds checking or error handling is performed
- The overflow over-writes the return address with a new address
- The new address is now pointing to different data than it should
- The new data contains malicious code
- The malicious code is executed with the privileges of the process that overflowed
- The exploit code now can run a virus, open a back door, format a drive, install a sniffer, or pretty much anything it wants to. Limited only by whatever privileges the user running the program has on that system.

Diagram:



How to use it?

Use of this exploit is almost ridiculously simple. To begin with download and install ICQ on a Windows computer. Next is obtain an ICQ UIN ID number, either by registering or using someone else's through one of the other exploits like spoofing. Searching on the Internet through any search engine, exploit databases like BugTraq or a 'how-to-hack' type web site will supply the attacker with instructions and sample code. Take the compiled assembler code of a malicious program, virus or backdoor program, copy the code using an editor and append it to the end of the URL. Log onto ICQ and compose a message, pasting the new URL and then send the message.

Signature of the attack:

Identifying this attack on the wire would be more than a little difficult. The traffic would not be very different from normal, if you allow ICQ in the first place. The only way to detect this specific attack would be the repetitive characters within the URL padding it in order to exceed the number that the receiving buffer can accept. I would expect that the majority of times this exploit would be run using the sample code from the post copied exactly. There may be some specific patterns within the assembler code appended to the URL, however they would change based on what was the intended malicious program to be run. After the attack had already succeeded the resulting backdoor traffic would likely tip you off that something may have happened.

How to protect against it?

The only obvious method of protection against this exploit would be not to expose yourself to it in the first place. Do not run ICQ on your own computer, do not allow it on your networks. Block ICQ ports at the firewalls, scan for ICQ on your clients, remove ICQ when found. The only other fix would be to never accept messages from ICQ users you do not know personally. Since it is not difficult to impersonate other ICQ users then the only other solution is to never click on URL's sent by any ICQ user.

Conclusion:

Reading about the many exploits and vulnerabilities in the ICQ client and protocol would definitely lead me to rethink allowing ICQ or similar programs on my computer or the networks I work with. It is relevant to note that the makers of ICQ themselves do not recommend using their product for 'mission critical' or 'content sensitive' types of communications. Quoting Yossi Vardi of the Isreali Mirabilis group "If you want to do something that will provide good security but will be palatable to a wide [number] of users, you have to see what you can do that will provide reasonable security, but will not create huge client." Doing some research for this paper I ran across this quote regarding ICQ by Allan Cox from December 1997 called 'ICQ so-called protocol'

"The ICQ protocol is ridiculously simplistic and riddled with security holes. So is the ICQ software. So ICQ users can be spoofed, have their machines crashed, or have evil haxxors run arbitrary code on their boxes. Geez, these poor users might as well run Internet Explorer!". ICQ was originally released in 1996 and has improved since then, however exploits continue to be found in both the protocols and client programs.

Sample Code:

Taken from the posting by Drew Copley to BugTraq. Available at:
http://www.securityfocus.com/templates/archive.pike?list=1&mid=41568
Copy the following characters and paste them into an ICQ message. The lines are wrapped by Word and should be continuous.

```
http://www.yahoo.com/sites.asp?^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ!!!!·P

!^Đ^Đ^Đ^Đ^Đ^Đ^Đ^Đ
```

Additional ICQ Vulnerabilities or Exploits:

- An attacker can easily 'spoof' the ICQ identifier, allowing them to send messages from any arbitrary ICQ user ID number.
- It is possible to hijack another users ICQ user ID number, change their password for ICQ, intercept and send messages appearing to be from the user. Usually the web service file retrieval vulnerability or ICQ password sniffer is used.
- You can 'sniff' the ICQ password, transmitted once per session, which can be used to hijack the users ICQ ID. Change their password, send and receive their messages.
- There are multiple methods to perform denial of service attacks, shutting down ICQ, their Internet dial up session, or crash Windows.
- Unauthorized remote access to the users hard drive is possible if they are running the ICQ Web service, allowing download of any file on their drives. The attacker can easily retrieve any file desired through the ICQ web server.
- While ICQ does have an option to 'hide' the client IP address, others can 'unhide' and view your IP address, even if you are in ICQ 'invisible' mode. Allowing an attacker to run other exploits, scans or denial of service attacks.
- ICQ can be used to flood a victim with hundreds of bogus messages, combined with an ICQ spoof from an arbitrary ICQ ID number. Effectively either rather annoying or a denial of service.
- One site reported that a downloadable ICQ password cracker executable is actually a Trojan horse back door. Users wanting to attack others through ICQ are unwittingly opening up their own systems to attack.
- System administrators, or attackers as well, can detect and locate ICQ clients on local networks or across the Internet. Either to shut them down or take advantage of one of these exploits.

There are no patches or fixes to most of the ICQ vulnerabilities or exploits from the vendor.

Bibliography:

root shell:

http://rootshell.com/secengine/search.cgi?query=icq

Term paper on ICQ protocol by Tom Ueltschi

http://omega.uta.edu/~tom/ICQ

ICQ Spoofing:

http://www.digivill.net/~minus/icq/

ICQ Protocol Specifications:

http://www.algonet.se/~henisak/icq/

ISS X-Force:

http://xforce.iss.net/static/3889.php

BugTraq:

http://www.securityfocus.com/archive/1/41568

http://www.securityfocus.com/bid/929

Article on Wired about ICQ:

http://www.wired.com/news/print/0,1294,12758,00.html

Posting regarding security of ICQ protocol by Alan Cox

http://www.insecure.org/sploits/icq.spoof.overflow.seq.html

packet storm:

http://packetstorm.securify.com/

CVE:

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0046

ICQ:

http://www.mirabilis.com/

http://www.icq.com/

http://www.icq.com/features/security/security-tutorial.html

SANS:

http://www.sans.org

Buffer Overflows:

http://www.nwfusion.com/newsletters/sec/0830sec1.html

Last Updated: July 24th, 2017

Upcoming Training

Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Memphis SEC504	Memphis, TN	Aug 21, 2017 - Aug 26, 2017	Community SANS
Mentor Session AW - SEC504	Milwaukee, WI	Aug 23, 2017 - Sep 29, 2017	Mentor
Mentor Session AW - SEC504	New York, NY	Aug 24, 2017 - Sep 08, 2017	Mentor
Mentor Session - SEC504	Denver, CO	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	SEC504 - 201709,	Sep 05, 2017 - Oct 12, 2017	vLive
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, Ireland	Sep 11, 2017 - Sep 16, 2017	Live Event
Mentor AW - SEC504	Santa Clara, CA	Sep 11, 2017 - Sep 22, 2017	Mentor
Mentor Session - SEC504	Arlington, VA	Sep 20, 2017 - Nov 01, 2017	Mentor
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, Netherlands	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Columbia SEC504	Columbia, MD	Sep 25, 2017 - Sep 30, 2017	Community SANS
Mentor Session - SEC504	Boston, MA	Sep 26, 2017 - Nov 07, 2017	Mentor
Mentor Session AW - SEC504	Houston, TX	Oct 02, 2017 - Dec 11, 2017	Mentor
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Mentor Session - SEC504	Columbia, SC	Oct 03, 2017 - Nov 14, 2017	Mentor
Community SANS Chicago SEC504	Chicago, IL	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event