扫雷辅助

- 窗口过程函数：`0x01001BC9`

-
  ```
  初级：0x209
  中级：0x20A
  高级：0x20B
  自定义：0x20C
  ```

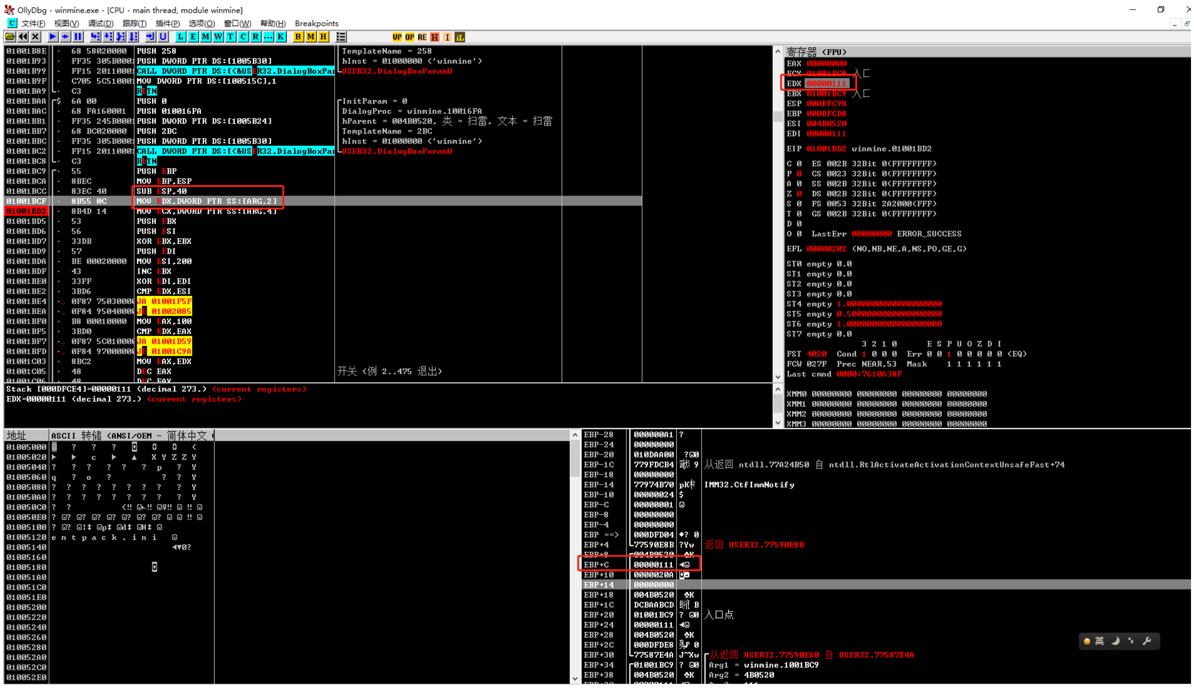
-
  ```
  雷个数基址：01005194
  dd 01005194
  ```

-
  ```
  雷区数组数据基地址：0x01005361
  0x8F 雷
  0x10 终止
  ```

-
  ```
  行数：0x01005338
  列数：0x01005334
  ```

-
  ```
  face坐标：250, 30
  第一个格子坐标：20, 60
  ```

菜单消息



OllyDbg - winmine.exe - [CPU - main thread, module winmine]

```
01001B8E  .  68 58020000     PUSH 258                                  ┌TemplateName = 258
01001B93  .  FF35 305B0001   PUSH DWORD PTR DS:[1005B30]               │hInst = 01000000 ('winmine')
01001B99  .  FF15 20110001   CALL DWORD PTR DS:[<&USER32.DialogBoxPar  └USER32.DialogBoxParamW
01001B9F  .  C705 5C510001   MOV DWORD PTR DS:[100515C],1
01001BA9  └. C3              RETN
01001BAA ┌$  6A 00           PUSH 0                                    ┌InitParam = 0
01001BAC  .  68 FA160001     PUSH 010016FA                             │DialogProc = winmine.10016FA
01001BB1  .  FF35 245B0001   PUSH DWORD PTR DS:[1005B24]               │hParent = 004B0520, 类 = 扫雷, 文本 = 扫雷
01001BB7  .  68 BC020000     PUSH 2BC                                  │TemplateName = 2BC
01001BBC  .  FF35 305B0001   PUSH DWORD PTR DS:[1005B30]               │hInst = 01000000 ('winmine')
01001BC2  .  FF15 20110001   CALL DWORD PTR DS:[<&USER32.DialogBoxPar  └USER32.DialogBoxParamW
01001BC8  └. C3              RETN
01001BC9 ┌.  55              PUSH EBP
01001BCA  .  8BEC            MOV EBP,ESP
01001BCC  .  83EC 40         SUB ESP,40
01001BCF  .  8B55 0C         MOV EDX,DWORD PTR SS:[ARG.2]
01001BD2  .  8B4D 14         MOV ECX,DWORD PTR SS:[ARG.4]
01001BD5  .  53              PUSH EBX
01001BD6  .  56              PUSH ESI
01001BD7  .  33DB            XOR EBX,EBX
01001BD9  .  57              PUSH EDI
01001BDA  .  BE 00020000     MOV ESI,200
01001BDF  .  43              INC EBX
01001BE0  .  33FF            XOR EDI,EDI
01001BE2  .  3BD6            CMP EDX,ESI
01001BE4  .  0F87 75030000   JA 01001F5F
01001BEA  .  0F84 95040000   JE 01002085
01001BF0  .  B8 00010000     MOV EAX,100
01001BF5  .  3BD0            CMP EDX,EAX
01001BF7  .  0F87 5C010000   JA 01001D59
01001BFD  .  0F84 97000000   JE 01001C9A
01001C03  .  8BC2            MOV EAX,EDX
01001C05  .  48              DEC EAX
                                                                       开关 (例 2..475 退出)
Stack [000DFCE4]=00000111 (decimal 273.) (current registers)
EDX=00000111 (decimal 273.) (current registers)
```

```
寄存器 (FPU)
EAX 00000000
ECX 01001BC9 入口
EDX 00000111
EBX 01001BC9 入口
ESP 000DFC98
EBP 000DFCD8
ESI 004B0520
EDI 00000111

EIP 01001BD2 winmine.01001BD2

C 0  ES 002B 32Bit 0(FFFFFFFF)
P 0  CS 0023 32Bit 0(FFFFFFFF)
A 0  SS 002B 32Bit 0(FFFFFFFF)
Z 0  DS 002B 32Bit 0(FFFFFFFF)
S 0  FS 0053 32Bit 2A2000(FFF)
T 0  GS 002B 32Bit 0(FFFFFFFF)
D 0
O 0  LastErr 00000000 ERROR_SUCCESS

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 1.0000000000000000000
ST5 empty 0.5000000000000000000
ST6 empty 1.0000000000000000000
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 4020  Cond 1 0 0 0  Err 0 0 1 0 0 0 0 0 (EQ)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
Last cmnd 0000:7610638F

XMM0 00000000 00000000 00000000 00000000
XMM1 00000000 00000000 00000000 00000000
XMM2 00000000 00000000 00000000 00000000
XMM3 00000000 00000000 00000000 00000000
```

```
EBP-28   000000A1 ?
EBP-24   00000000
EBP-20   010DAA00  ?
EBP-1C   779FDCB4     从返回 ntdll.77A24B50 自 ntdll.RtlActivateActivationContextUnsafeFast+74
EBP-18   00000000
EBP-14   77974B70 pK   IMM32.CtfImmNotify
EBP-10   00000024 $
EBP-C    00000001
EBP-8    00000000
EBP-4    00000000
EBP ==>  000DFD04  ? 0
EBP+4   └77590E8B ?Yw  返回 USER32.77590E8B
EBP+8   ┌004B0520
EBP+C    00000111
EBP+10   0000020A
EBP+14   00000000
EBP+18   004B0520  K
EBP+1C   DCBAABCD   B
EBP+20   01001BC9 ?    入口点
EBP+24   00000111
EBP+28   004B0520  K
EBP+2C   000DFDE8   0
EBP+30  └77587E4A J^Xw ┌从返回 USER32.77590E60 自 USER32.77587E4A
EBP+34  ┌01001BC9 ?    │Arg1 = winmine.1001BC9
EBP+38   004B0520  K   │Arg2 = 4B0520
```

```
以高级为例（注入器注入）：
push 0
push 0x20B
push 0x111
push 0xD0ADC
call 0x01001BC9
```

代码展示

```
HWND hwnd;
RECT rect;
DWORD pid;
HANDLE process_hwnd;

void init() {
    hwnd = ::FindWindowA(NULL, "扫雷");
    ::GetWindowThreadProcessId(hwnd, &pid);
    process_hwnd = ::OpenProcess(PROCESS_ALL_ACCESS, NULL, pid);
    ::GetWindowRect(hwnd, &rect);
}


void CWinMineDlg::OnBnClickedBtnCj()
{
    init();
    if (!hwnd) {
        ::MessageBoxA(NULL, "扫雷游戏未打开", "错误", MB_OK);
        return;
    }
    ::SendMessageA(hwnd, WM_COMMAND, 0x209, 0);
    m_edit_show_data.Empty();
    UpdateData(FALSE);
}


void CWinMineDlg::OnBnClickedBtnZj()
{
    init();
    if (!hwnd) {
        ::MessageBoxA(NULL, "扫雷游戏未打开", "错误", MB_OK);
        return;
    }
    ::SendMessageA(hwnd, WM_COMMAND, 0x20A, 0);
    m_edit_show_data.Empty();
    UpdateData(FALSE);
}


void CWinMineDlg::OnBnClickedBtnGj()
{
    init();
    if (!hwnd) {
        ::MessageBoxA(NULL, "扫雷游戏未打开", "错误", MB_OK);
        return;
    }
    ::SendMessageA(hwnd, WM_COMMAND, 0x20B, 0);
    m_edit_show_data.Empty();
    UpdateData(FALSE);
}


void CWinMineDlg::OnBnClickedBtnZdy()
{
    init();
    if (!hwnd) {
        ::MessageBoxA(NULL, "扫雷游戏未打开", "错误", MB_OK);
        return;
    }
    ::SendMessageA(hwnd, WM_COMMAND, 0x20C, 0);
    m_edit_show_data.Empty();
    UpdateData(FALSE);
}

int tx, ty, pos_x, pos_y;
POINT pre_pos;
void CWinMineDlg::OnBnClickedBtnReadChess()
{
    m_edit_show_data.Empty();
    init();
    if (!hwnd) {
        ::MessageBoxA(NULL, "扫雷游戏未打开", "错误", MB_OK);
        return;
    }

    // 设置窗口置顶
    ::SetWindowPos(hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE | SWP_NOSIZE | SWP_SHOWWINDOW);

    ::GetCursorPos(&pre_pos);

    int face_x = rect.left + (rect.right - rect.left) / 2;
    int face_y = rect.top + 75;

    ::SetCursorPos(face_x, face_y);
    ::Sleep(10);
    mouse_event(MOUSEEVENTF_LEFTDOWN | MOUSEEVENTF_LEFTUP, 0, 0, 0, 0);
    ::Sleep(10);

    ::SetCursorPos(rect.left + 20, rect.top + 110);
    ::Sleep(10);
    mouse_event(MOUSEEVENTF_LEFTDOWN | MOUSEEVENTF_LEFTUP, 0, 0, 0, 0);
    ::Sleep(10);

    ::SetCursorPos(pre_pos.x, pre_pos.y);

    ::Sleep(1000);

    unsigned char chess_data[24][32] = {};
    ::ReadProcessMemory(process_hwnd, (LPCVOID)0x01005361, (LPVOID)&chess_data, 24 * 32, NULL);

    int irow;
    ::ReadProcessMemory(process_hwnd, (LPCVOID)0x01005338, (LPVOID)&irow, 4, NULL);

    CString tstr;
    for (int i = 0; i < irow; i++) {
        for (int j = 0; j < 32; j++) {
            if (0x10 == chess_data[i][j]) break;
            tstr.Format("%02X ", chess_data[i][j]);
            m_edit_show_data += tstr;
        }
        m_edit_show_data += "\r\n";
    }

    UpdateData(FALSE);
}

void MoveToAndLeftClick(HWND hwnd, int x, int y) {
    int lparam = (pos_y << 16) + pos_x;
    ::SendMessageA(hwnd, WM_LBUTTONDOWN, MK_LBUTTON, lparam);
    ::SendMessageA(hwnd, WM_LBUTTONUP, 0, lparam);
}

void CWinMineDlg::OnBnClickedBtnAutoClear()
{
    m_edit_show_data.Empty();
    init();
    if (!hwnd) {
        ::MessageBoxA(NULL, "扫雷游戏未打开", "错误", MB_OK);
        return;
    }

    unsigned char chess_data[24][32] = {};
    ::ReadProcessMemory(process_hwnd, (LPCVOID)0x01005361, (LPVOID)&chess_data, 24 * 32, NULL);

    int irow;
    ::ReadProcessMemory(process_hwnd, (LPCVOID)0x01005338, (LPVOID)&irow, 4, NULL);

    tx = 20, ty = 60;
    for (int i = 0; i < irow; i++) {
        for (int j = 0; j < 32; j++) {
            if (0x10 == chess_data[i][j]) break;
            pos_x = tx + j * 16;
            pos_y = ty + i * 16;

            if (0x8f != chess_data[i][j])
                MoveToAndLeftClick(hwnd, pos_x, pos_y);
        }
    }
}
```