

## LyScriptUtils 转换工具包

该工具包其目的是辅助LyScript插件实现进制与字符串或字节序列的快速转换，协助逆向工作者更好的反汇编，工具包默认支持32位与64位环境。

- 安装进制转换工具包: `pip install LyScriptUtils`

将一个特定字节切割成等量的字节数组

```
from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 切割32位
    # [18, 52, 86, 120]
    ref = split_int32(0x12345678)
    print(ref)

    # 切割64位
    # [0, 255, 255, 255, 18, 52, 86, 120]
    ref = split_int64(0x0FFFFFFF12345678)
    print(ref)

    # 自定义切割字节数
    # [18, 52]
    ref = split_int_bits(16, 0x1234)
    print(ref)
```

将一个十六进制字符串转成等量的INT类型

```
from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 0x1234
    ref = str2int16("\x12\x34\x56")
    print(hex(ref))

    # 0x12345678
    ref = str2int32("\x12\x34\x56\x78")
    print(hex(ref))

    # 0x121c5678121c5678
    ref = str2int64("\x12\x34\x56\x78\x12\x34\x56\x78")
    print(hex(ref))

    # 0x1234
    ref = nstr2halfword("\x12\x34")
    print(hex(ref))

    # 0x12345678121c567812345678121c5678
```

```

    ref =
str2int_bits(128, "\x12\x34\x56\x78\x12\x34\x56\x78\x12\x34\x56\x78\x12\x34\x56\x78")
    print(hex(ref))

```

将一个十六进制字符串转成等量的INT类型,并反转参数

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 0x3412
    ref = str2int16_swapped("\x12\x34\x56")
    print(hex(ref))

    # 0x12563412
    ref = str2int32_swapped("\x12\x34\x56\x12\x34\x56")
    print(hex(ref))

    # 0x3412563412563412
    ref = str2int64_swapped("\x12\x34\x56\x12\x34\x56\x12\x34\x56\x12\x34\x56")
    print(hex(ref))

    # 0x3412
    ref = str2int_bits_swapped(16, "\x12\x34")
    print(hex(ref))

```

将字符串转换成对小端序或大端序的十六进制整数

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 小端序
    # 0x78563412
    ref = str2littleendian("\x12\x34\x56\x78")
    print(hex(ref))

    # 0x78563412
    ref = intel_str2int("\x12\x34\x56\x78")
    print(hex(ref))

    # 0x78563412
    ref = intel_str2int("\x12\x34\x56\x78")
    print(hex(ref))

    # 大端序
    # 0x12345678
    ref = str2int32("\x12\x34\x56\x78")
    print(hex(ref))

    # 0x12345678
    ref = str2bigendian("\x12\x34\x56\x78")
    print(hex(ref))

```

将一个十六进制整数转换为字节序列

```
from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    ref = int2str16(0x1234)
    if ref == '\x12\x34':
        print("int2str16")

    ref = halfword2bstr(0x1234)
    if ref == '\x12\x34':
        print("int2str16")

    ref = short2bigstr(0x1234)
    if ref == '\x12\x34':
        print("int2str16")

    ref = big_short(0x1234)
    if ref == '\x12\x34':
        print("int2str16")
```

将一个十六进制整数转换为字节序列,并反转参数

```
from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    ref = int2str16_swapped(0x1234)
    if ref == '\x34\x12':
        print("int2str16_swapped")

    ref = halfword2istr(0x1234)
    if ref == '\x34\x12':
        print("halfword2istr")

    ref = intel_short(0x1234)
    if ref == '\x34\x12':
        print("intel_short")

    ref = intel_short(0x123445678)
    if ref == '\x78\x56':
        print("intel_short")
```

将INT类型十六进制数值,转换为str32位字节序列

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    ref = int2str32(0x12345678)
    if ref == '\x12\x34\x56\x78':
        print("int2str32")

    ref = big_order(0x12345678)
    if ref == '\x12\x34\x56\x78':
        print("big_order")

```

将一个INT类型,转换为str32字节序列,并反转参数

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    ref = int2str32_swapped(0x12345678)
    if ref == '\x78\x56\x34\x12':
        print("int2str32_swapped")

    ref = intel_order(0x12345678)
    if ref == '\x78\x56\x34\x12':
        print("intel_order")

```

将一个十六进制INT整数,转换成一个二进制字符串

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 00010010001101000101011001111000
    ref = print_binary(0x12345678)
    if ref == '00010010001101000101011001111000':
        print(ref)

    # 0101011001111000
    ref = binary_string_short(0x12345678)
    if ref == '0101011001111000':
        print(ref)

```

针对有符号数与无符号数的类型转换函数

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 无符号数转换
    if uint16(0xffff) == 0xffff:
        print("uint16")

    if uint64(0x0000000fffffffff) == 0x0000000fffffffff:

```

```

    print("uint64")

# 有符号数转换
if sint16(0xffff) == -1:
    print("sint16")

if sint16(0xffff) == sint16(-1):
    print("sint16==sint16")

if signedshort(0xffff) == -1:
    print("signedshort")

# 有符号大整数
if sint32(-1) == -1:
    print("sint32")

if big2int(0x123456789) == 0x23456789:
    print("big2int")

```

将一个无符号数格式化为十六进制字符串

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    if(uintfmt_bits(32,0x12345678)=='0x12345678'):
        print("uintfmt_bits")

    if(uintfmt_bits(16,0x1234)=='0x1234'):
        print("uintfmt_bits")

    if(uint16fmt(0x123456)=='0x3456'):
        print("uintfmt_bits")

    if(uint16fmt(-0x123456)=='0xcbaa'):
        print("uintfmt_bits")

    if(uint32fmt(0x1234)=='0x00001234'):
        print("uintfmt_bits")

    if(uint64fmt(0x12345678)=='0x0000000012345678'):
        print("uintfmt_bits")

    if(uint64fmt(-1)=='0xffffffffffffffff'):
        print("uintfmt_bits")

    if(uint8fmt(0x0f)=='0x0f'):
        print("uint8fmt")

```

将一个有符号数格式化为十六进制字符串

```

from LyScript32 import MyDebug
from LyScriptUtils import *

```

```

if __name__ == '__main__':
    # sint16fmt
    if(sint16fmt(0x1234)=='0x1234'):
        print("sint16fmt")

    if(sint16fmt(-0x1234)=='-0x1234'):
        print("sint16fmt")

    # sint32fmt
    if(sint32fmt(-0x1234)=='-0x00001234'):
        print("sint32fmt")

    # sint64fmt
    if(sint64fmt(-1)=='-0x0000000000000001'):
        print("sint64fmt")

```

将一个十六进制INT整数反转

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 0x78563412
    if(byteswap_32(0x12345678)==0x78563412):
        print("32位反转")

    # 0x5634129078563412
    if(byteswap_64(0x1234567890123456)==0x5634129078563412):
        print("64位反转")

```

对一个字符串执行十六进制格式化输出

```

from LyScript32 import MyDebug
from LyScriptUtils import *

if __name__ == '__main__':
    # 十六进制格式化输出
    # [0x12][0x34]
    hexpr = hexprint(''.join(int2list(uint16(0x1234))[2:4]))
    print(hexpr)

    # 字节序列转换成字符串
    # [8b][ec][12][ff][8b][ec][12][ff]
    byte_string = prettyprint("\x8b\xec\x12\xff\x8b\xec\x12\xff")
    print(byte_string)

    # 字符串转换成shellcode
    # unsigned char buf[] = "\x6c\x79\x73\x68\x61\x72\x6b"; // 7 bytes
    shellcode = c_array("lyshark")
    print(shellcode)

    # dump shellcode
    dump = shellcode_dump("\x6c\x79\x73\x68\x61\x72\x6b")

```

```
# 将字符串转为二进制数组
# [0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1]
ref = binary_from_string("lushark")
print(ref)

# 输出字符串的十六进制格式
# [('6c 79 73 68 61 72 6b ', 'lyshark')]
ref = hexdump("lyshark")
print(ref)
```