hedgehog-tools/nighthawk_str_decoder.py at main · struppigel/hedgehog-tools · GitHub

github.com/struppigel/hedgehog-tools/blob/main/nighthawk_str_decoder.py

struppigel

struppigel/
hedgehog-tools

2 Contributors | 0 Issues | 26 Stars | 2 Forks

```
import idautils

# Author: Karsten Hahn @ GDATA CyberDefense
# Twitter: @struppigel
# Mastodon: struppigel@infosec.exchange
# IDAPython Script to decode NightHawk strings
# sha256:
9a57919cc5c194e28acd62719487c563a8f0ef1205b65adbe535386e34e418b8
# sha256:
0551ca07f05c2a8278229c1dc651a2b1273a39914857231b075733753cb2b988

def find_cipher_addr():
# 33 ED xor ebp, ebp
# 48 39 6B 10 cmp [rbx+10h], rbp
```

```
    # 76 4E jbe short loc_1800456B8
    # 33 F6 xor esi, esi
    # 4C 8D 3D ?? ?? ?? ?? lea r15, alphabetString
    address_found = ida_search.find_binary(0, 0xffffffffffffffff, "33 ED 48 39 6B 10 76 4E 33 F6 4C 8D 3D", 0, 0)
    cmd_start = 10
    cipher_alphabet_addr = idc.get_operand_value(address_found + cmd_start,1)
    cipher_alphabet = idc.get_strlit_contents(cipher_alphabet_addr, -1, 0).decode("utf-8")
    print("extracted alphabet:", cipher_alphabet)
    return cipher_alphabet

    def find_plain_addr():
    # 49 8B CF mov rcx, r15 ; Str
    # E8 ?? ?? ?? ?? call strchr
    # 48 85 C0 test rax, rax
    # 74 1D jz short loc_7FFACE4089C6
    # 49 2B C7 sub rax, r15
    # 48 8D 0D ?? ?? ?? ?? lea rcx, substitution_alphabet2
    address_found = ida_search.find_binary(0, 0xffffffffffffffff, "49 8B CF E8 ? ? ? ? 48 85 C0 74 1D 49 2B C7 48 8D 0D", 0, 0)
    cmd_start = 16
    plain_addr = idc.get_operand_value(address_found + cmd_start,1)
    plain_alphabet = idc.get_strlit_contents(plain_addr, -1, 0).decode("utf-8")
    print("extracted alphabet:", plain_alphabet)
    return plain_alphabet

    def decryption_func_addr():
    # 48 8B C4 mov rax, rsp
```

```
# 48 89 58 18 mov [rax+18h], rbx
# 48 89 68 20 mov [rax+20h], rbp
# 48 89 50 10 mov [rax+10h], rdx
# 48 89 48 08 mov [rax+8], rcx
# 56 push rsi
# 57 push rdi
# 41 57 push r15
# 48 83 EC 30 sub rsp, 30h
# 48 8B DA mov rbx, rdx
# 48 8B F9 mov rdi, rcx
# 83 60 D8 00 and dword ptr [rax-28h], 0
# 48 8B CA mov rcx, rdx
# 48 83 7A 18 10 cmp qword ptr [rdx+18h], 10h
# 72 03 jb short loc_7FFACE40896A
# 48 8B 0A mov rcx, [rdx] ; Str
return ida_search.find_binary(0, 0xffffffffffffffff, "48 8B C4 48 89 58 18 48 89 68 20 48 89 50 10 48 89 48 08 56 57 41 57 48 83 EC 30 48 8B DA 48 8B F9 83 60 D8 00 48 8B CA 48 83 7A 18 10 72 03 48 8B 0A", 0, 0)

def decode_string(encoded, cipher_alphabet, plaintext_alphabet):
decoded = []
for character in encoded:
if character in cipher_alphabet:
decoded.append(plaintext_alphabet[cipher_alphabet.find(character)])
else:
decoded.append(character)
return "".join(decoded)
```

```
if __name__ == "__main__":

cipher_alphabet = find_cipher_addr()
plaintext_alphabet = find_plain_addr()

for ref in idautils.XrefsTo(decryption_func_addr(),0):
caller_addr = ref.frm
for h in Heads(caller_addr - 0x20, caller_addr):
if print_insn_mnem(h) == "lea":
if idc.get_operand_type(h,1) == 2:
string_addr = idc.get_operand_value(h,1)
encoded_str = idc.get_strlit_contents(string_addr, -1, 0)
decoded_str = decode_string(encoded_str.decode("utf-8"), cipher_alphabet,
plaintext_alphabet)
print(decoded_str, 'at', hex(string_addr))
idc.set_cmt(h, decoded_str, True)
idc.set_cmt(caller_addr, decoded_str, True)
idc.set_cmt(string_addr, decoded_str, True)
break
```