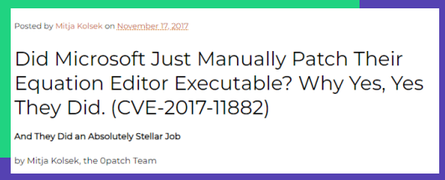
BINARY NINJA

INTRODUCTION

Decompilation is the process of turning compiled binaries back into code. It is used for many things – malware analysis, fixing bugs, porting old software to new platforms, and much more.



Posted by Mitja Kolsek on November 17, 2017

Did Microsoft Just Manually Patch Their Equation Editor Executable? Why Yes, Yes They Did. (CVE-2017-11882)

And They Did an Absolutely Stellar Job

by Mitja Kolsek, the 0patch Team

INFORMATION LOSS

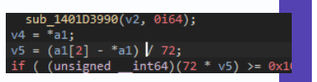
Compilers remove information when compiling code such as variable names etc. Decompilers must work without this and still produce clean output.

DIVISION-BY-MULTIPLICATION

CPUs divide very slowly and multiply quickly. As a result, compilers optimize divisions by performing a multiplication operation instead. The operation on the right is a division by 288 - yet the compiler transforms it into a multiplication by a strange number. This is because it can take the lower bits only to divide rapidly by a large number – thus preserving accuracy and maintaining performance. Decompilers don't always handle this correctly – on the right is a correct decompilation and on the left is an incorrect one.

```
mov     rax, 0xe38e38e38e38e39
imul    rcx
```

```
int64_t rdx_1;
rdx_1 = HIGHQ((0xe38e38e38e38e39 * (arg1[2] - r8_2)));
rax_1 = LOWQ((0xe38e38e38e38e39 * (arg1[2] - r8_2)));
int64_t rdx_2 = (rdx_1 >> 2);
if (((rdx_2 + (rdx_2 >> 0x3f)) * 0x48) >= 0x1000)
{
```



```
sub_1401D3990(v2, 0i64);
v4 = *a1;
v5 = (a1[2] - *a1) / 72;
if ( (unsigned __int64)(72 * v5) >= 0x1
```

BINARY NINJA

Binary ninja is a decompiler that is partially open source, partially commercial. It is improving rapidly, however it still has flaws with the decompilation process.

FIXING THE BUG

TODO – not done with this yet

CONCLUSION