KEYPATCH: binary patcher for IDA Pro

http://keystone-engine.org/keypatch

NGUYEN Anh Quynh <aquynh -at- gmail.com>

Trada hacking - 16/9/2016
Who am I

- **Nguyen Anh Quynh, aqquynh -at- gmail.com**
  - Nanyang Technological University, Singapore
  - PhD in Computer Science
  - Operating System, Virtual Machine, Binary analysis, etc
  - Capstone disassembler: [http://capstone-engine.org](http://capstone-engine.org)
  - Unicorn emulator: [http://unicorn-engine.org](http://unicorn-engine.org)
  - Keystone assembler: [http://keystone-engine.org](http://keystone-engine.org)
Binary patching

- CrackMe, CTF challenges
- Malware analysis
- Modify binary without source code :-}
URLZone Banking Trojan

```
00414ef2  ffd1    call    ecx
00414ef4  83f800  cmp      eax, 0
00414ef7  7408    je       image00400000+0x14f01 (00414f01)
00414ef9  6a00    push     0
00414efb  ffd3    call    ebx {ntdll!RtlExitUserThread (77a5f608)}
00414efd  a1fcbacc59  mov    eax, dword ptr ds:[59CCBAFCh]
00414f02  5b      pop      ebx
```
IDA Pro

- [https://www.hex-rays.com](https://www.hex-rays.com)
- De-facto binary analysis tool
- Extendable with plugin SDK (C, Python)
Built-in binary patcher of IDA

- Modify binary code with menu "Edit | Patch program | Assemble..."

- Save changes permanently to binary file
  - Menu "Edit | Patch program | Apply patches to input file..."
How it work?

Input Assembly
(GUI editor)

New assembly

Assembler
(compile assembly)

New code

Patcher
(write binary to file)

Modified binary
Problems of IDA built-in binary patcher

Input Assembly (GUI editor)

New assembly

Assembler (compile assembly)

New code

Patcher (write binary to file)

Modified binary

- Preview new code?
- Padding NOP for orphan bytes?
- Logging changes?
- Fill a range?
- Revert (Undo) modification?

- Handle nothing but X86 input
- X86 support is very poor :-(

KEYPATCH: binary patcher for IDA Pro
Keypatch Solution

Input Assembly (GUI editor)

New assembly

Assembler (compile assembly)

New code

Patcher (write binary to file)

Modified binary
Keystone == Next Generation Assembler Framework
Assembler framework

**Definition**
- Compile assembly instructions & returns encoding as sequence of bytes
  - Ex: `inc EAX → 40`
- May support high-level concepts such as macro, function, etc
- Framework to build apps on top of it

**Applications**
- Dynamic machine code generation
  - Binary rewrite
  - Binary searching
Good assembler framework?

- True framework
  - Embedded into tool without resorting to external process
- Multi-arch
  - X86, Arm, Arm64, Mips, PowerPC, Sparc, etc
- Updated
  - Keep up with latest CPU extensions
- Multi-platform
  - *nix, Windows, Android, iOS, etc
- Bindings
  - Python, Ruby, Go, NodeJS, etc
Existing assembler frameworks

- Nothing is up to our standard, even in 2016!
  - Yasm: X86 only, no longer updated
  - Intel XED: X86 only, miss many instructions & closed-source
  - Other important archs: Arm, Arm64, Mips, PPC, Sparc, etc?
Life without assembler frameworks?

- People are very much struggling for years!
  - Use existing assembler tool to compile assembly from file
  - Call linker to link generated object file
  - Use executable parser (ELF) to parse resulted file for final encoding
- Ugly and inefficient
- Little control on the internal process & output
- Cross-platform support is very poor
"If not now, then when? If not you, then who?" - Kailash Satyarthi
Keystone (architecture)

From Wikipedia, the free encyclopedia

This article is about the architectural element. For other uses, see Keystone (disambiguation).

A **keystone** is the wedge-shaped stone piece at the **apex** of a masonry **arch**, the generally round one at the apex of a **vault**. In both cases it is the final piece placed during construction and locks all the stones into position, allowing the arch or vault to bear weight.\[1\][2][3] In both arches and vaults, keystones are often enlarged beyond the structural requirements, and often decorated in some way. Keystones are often placed in the centre of the flat top of openings such as doors and windows, essentially for decorative effect.
Timeline

- Indiegogo campaign started on March 17th, 2016 (for 3 weeks)
  - 99 contributors, 4 project sponsors
- Beta code released to beta testers on April, 2016
  - Only Python binding available at this time
- Version 0.9 released on May, 2016: http://keystone-engine.org
  - More bindings by beta testers: NodeJS, Ruby, Go & Rust
- Version 0.9.1 released on July 27th, 2016
  - 2 more bindings: Haskell & OCaml
Keystone engine

- True framework
  - Embedded into tool without resorting to external process
- Multi-arch
  - X86, Arm, Arm64, Mips, PowerPC, Sparc, Hexagon, SystemZ
- Updated
  - Keep up with latest CPU extensions
- Multi-platform
  - *nix, Windows, Android, iOS, etc
- C++ core & multi-bindings
  - Python, Ruby, Go, NodeJS, OCaml, Rust, Haskell
- Support various X86 undocumented instructions
- Compact & lightweight: 10× smaller than LLVM
Keypatch binary patcher for IDA
Keypatch

- Co-developed with Thanh Nguyen (VNSecurity.net)
- Open source IDA plugin http://keystone-engine.org/keypatch
- Tool for assembling & patching in IDA
- Built on top of Keystone assembler framework
  - Version 1.0 released at BlackHat USA 2016, August 4th, 2016
  - Version 2.0 released on September 14th, 2016
  - Version 2.0.1 released on September 15th, 2016
Keypatch - Patcher

Syntax: Intel
Address: .text:00000000000015909
Original: jz short loc_158F6
- Encode: 74 EB
- Size: 2
Assembly: xor eax, eax
- Fixup: xor eax, eax
- Encode: 31 C0
- Size: 2

- NOPs padding until next instruction boundary
- Save original instructions in IDA comment

Cancel Patch
Keypatch - Fill Range

Syntax: Intel
Start: .text:000000000001594B
End: .text:0000000000015950
Size: 5

Assembly:
- Fixup: nop
- Encode: 90
- Size: 1

- NOPs padding until next instruction boundary
- Save original instructions in IDA comment

Cancel   Patch
Keypatch vs IDA’s built-in patcher

- More friendly
  - Code preview
  - Padding NOPs automatically
  - Logging modifications
  - Fill a range of selected code
  - Assembler (do not modify)
  - Revert (undo)

- Support 8 architectures
  - Arm, Arm64, Hexagon, Mips, PowerPC, Sparc, SystemZ, X86
  - X86 support is fantastic

- Open source
Conclusions

- **Keypatch** is a superior binary patcher for IDA
  - Multi-arch + multi-platform
  - Feature-rich & friendly
  - Open source

- Looking for new contributors for our open source projects
  - Keypatch + Keystone engine
  - Capstone engine + Unicorn engine
References

- Keypatch: http://keystone-engine.org/keypatch
- Keystone assembler
  - Homepage: http://keystone-engine.org
  - Twitter: @keystone_engine
  - Github: http://github.com/keystone-engine/keystone
  - Mailing list: http://freelists.org/list/keystone-engine
Questions and answers

KEYPATCH: binary patcher for IDA Pro

http://keystone-engine.org/keypatch

NGUYEN Anh Quynh <aquynh -at- gmail.com>