Lecture 02: Foundations

August 26, 2024

What is Reverse Engineering?

1

2

3

What is Reverse Engineering?

engineering the art or science of making practical application of the knowledge of pure sciences, as physics or chemistry, as in the construction of engines, bridges, buildings, mines, ships, and chemical plants[1].

[1] Dictionary.com Unabridged.

2

3

What is Reverse Engineering?

engineering the art or science of making practical application of the knowledge of pure sciences, as physics or chemistry, as in the construction of engines, bridges, buildings, mines, ships, and chemical plants[1].

software engineering a systematic approach to the analysis, design, implementation and maintenance of software[2].

[1] Dictionary.com Unabridged.

[2] The Free On-line Dictionary of Computing

What is Reverse Engineering?

engineering the art or science of making practical application of the knowledge of pure sciences, as physics or chemistry, as in the construction of engines, bridges, buildings, mines, ships, and chemical plants[1].

software engineering a systematic approach to the analysis, design, implementation and maintenance of software[2].

reverse engineering the process of extracting the knowledge or design blueprints from anything man-made[3].

---

[1] Dictionary.com Unabridged.

[2] The Free On-line Dictionary of Computing

[3] Textbook

What is Reverse Engineering?

engineering the art or science of making practical application of the knowledge of pure sciences, as physics or chemistry, as in the construction of engines, bridges, buildings, mines, ships, and chemical plants[1].

software engineering a systematic approach to the analysis, design, implementation and maintenance of software[2].

reverse engineering the process of extracting the knowledge or design blueprints from anything man-made[3].

software reverse engineering the process of determining the purpose, design, structure, and implementation of software systems.

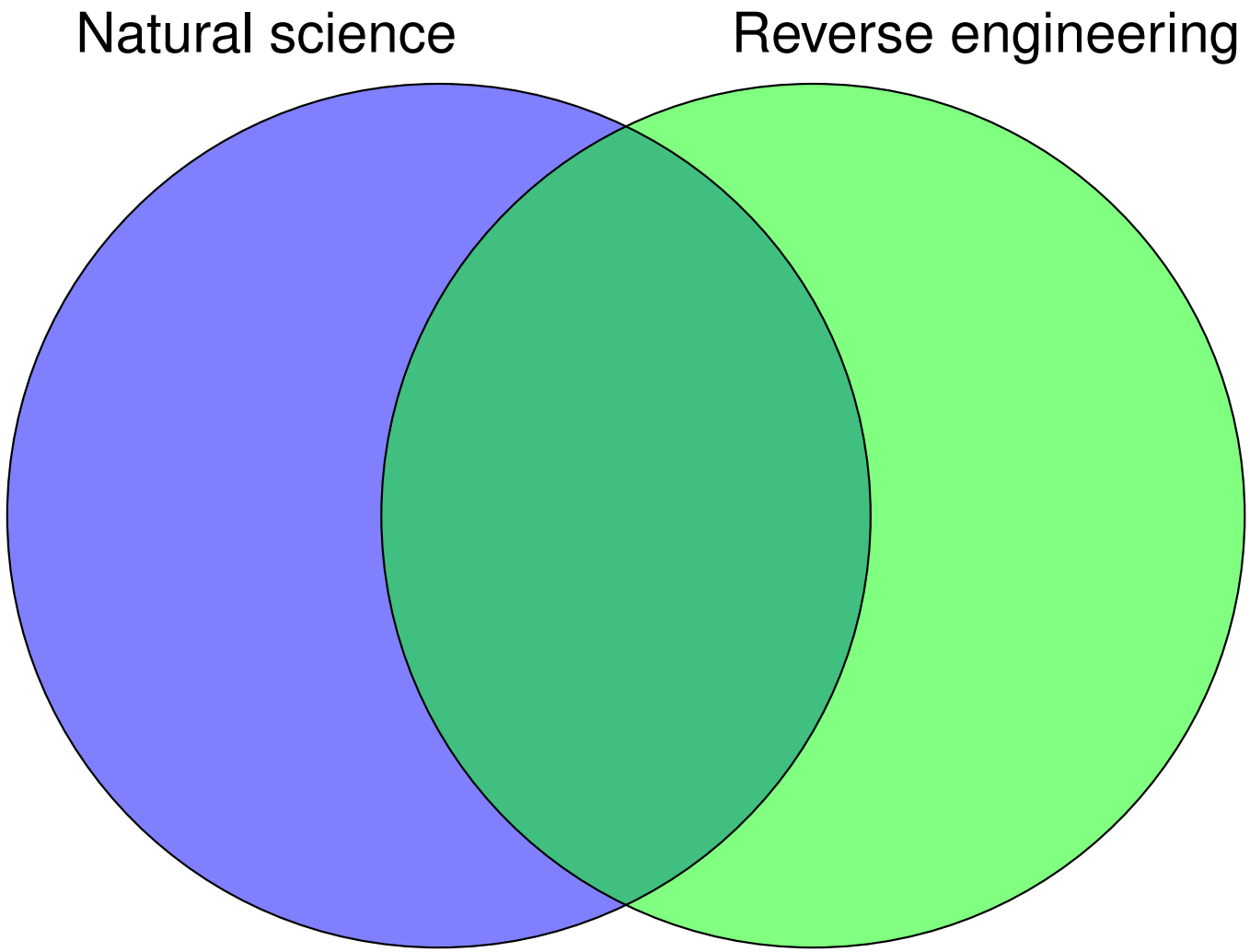
---

[1] Dictionary.com Unabridged.

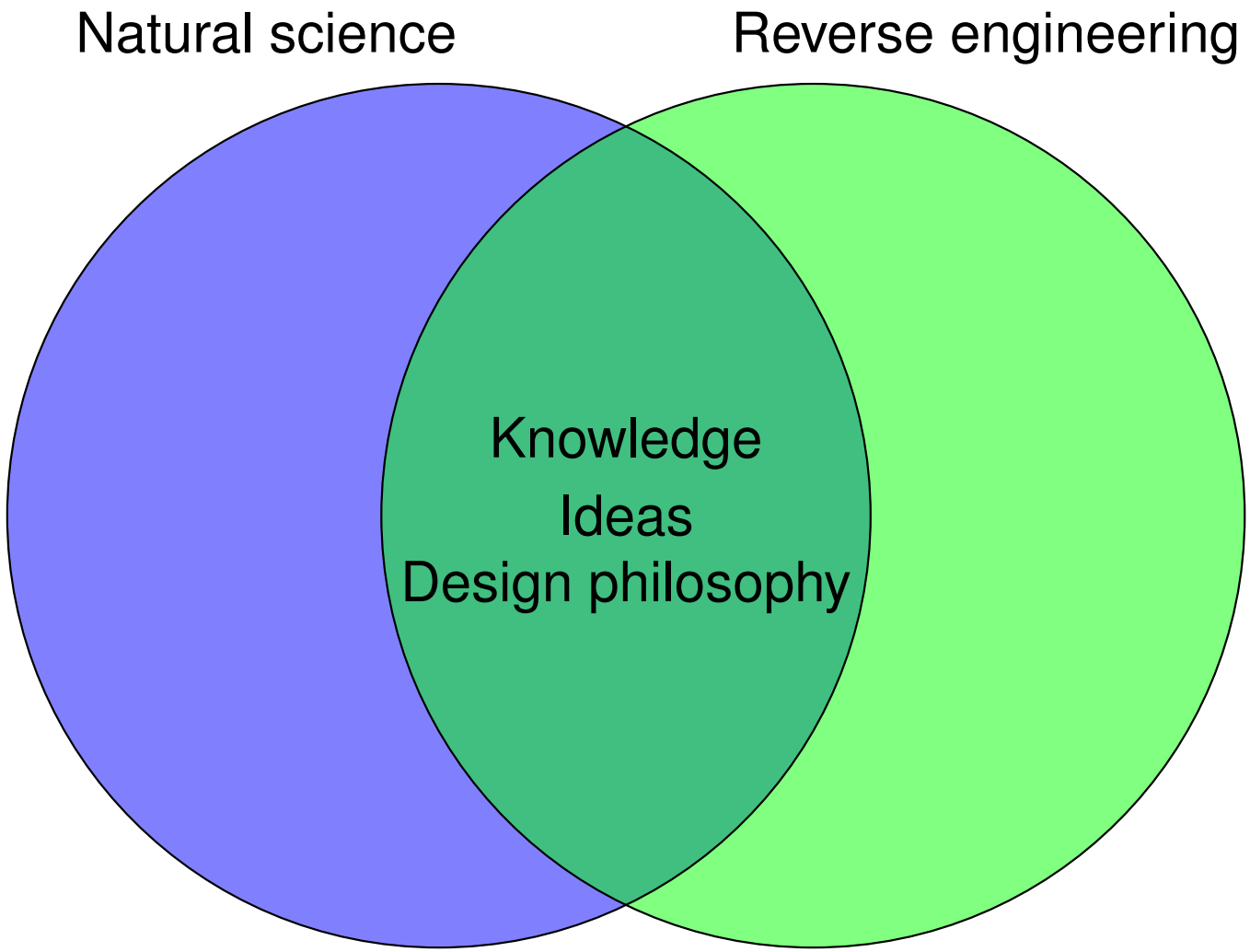
[2] The Free On-line Dictionary of Computing

[3] Textbook

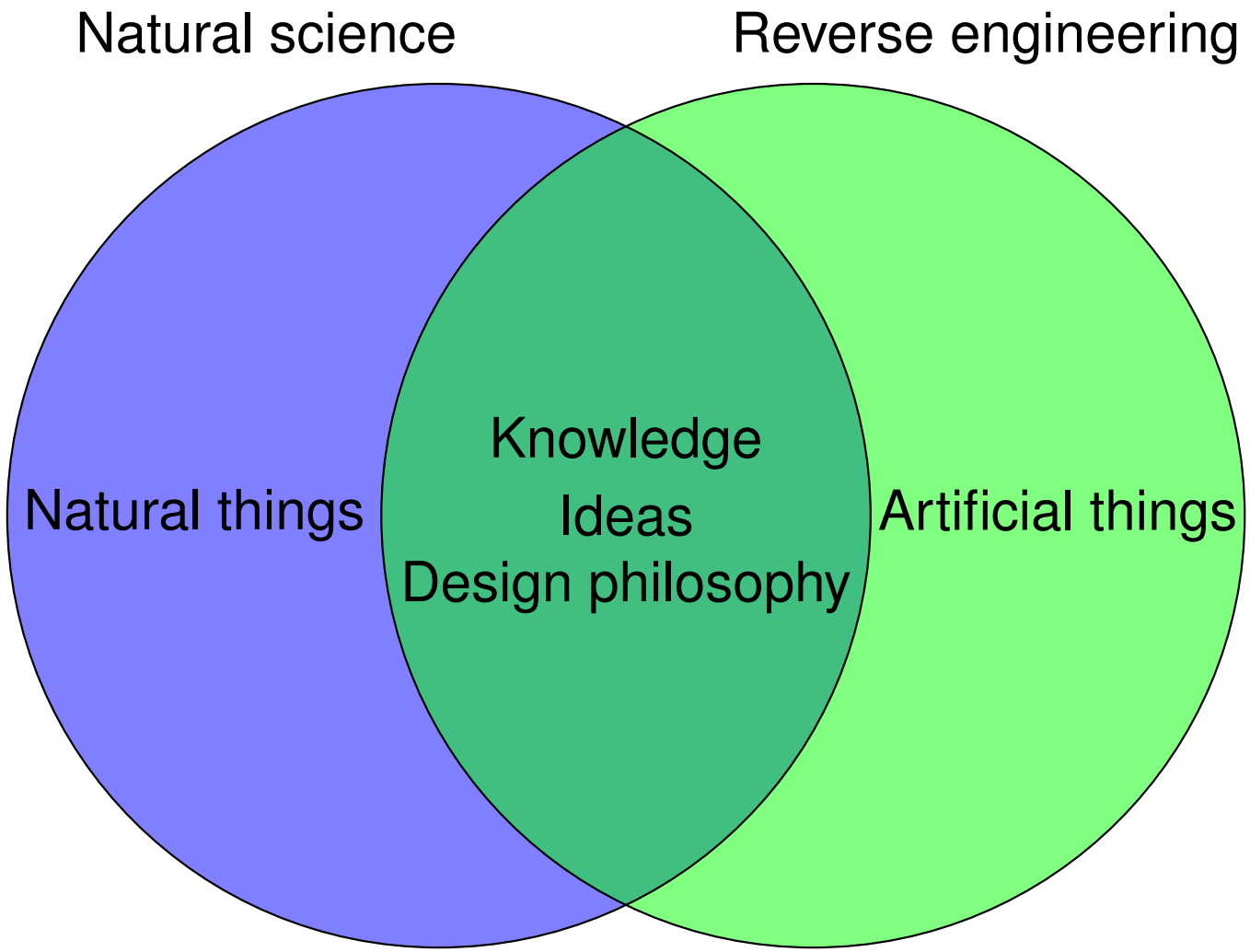
Comparison with natural science



Comparison with natural science



Comparison with natural science



Applications

Why reverse engineer?

Applications

Why reverse engineer?

Computer Security

Software development

Applications

Why reverse engineer?

Computer Security

- Understand cryptographic algorithms
- Analyze malicious software
- Defeat digital rights management (DRM)
- Audit software security

Software development

- Develop interoperable software
- Evaluate software quality
- Develop competing software

Low-level Software

Assembly language the lowest software level programming language for a particular platform; the mnemonic expression of machine code; the language of reverse engineering.

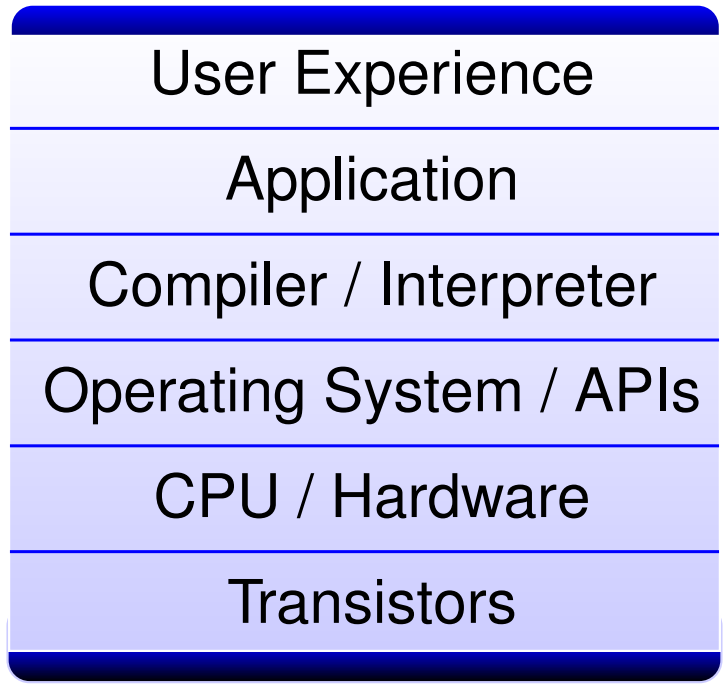
Compiler translates a program in a high-level language (e.g., C++) into machine code for a target platform.

Virtual machine a machine abstraction which executes bytecodes (a higher-level assembly) in order to achieve platform independence (e.g., Java, .NET)

Operating system a program which manages computer resources and performs operations on behalf of user applications.

Low-level Software

Layered Model



Low-level sofware

Assembly

- Lowest level in software
- Anything software does is visible here
- “Language of reversing”

- A class of languages
- Representation of machine code
- Opcodes & Operands
- x86 & amd64/x64/x86_64

Low-level sofware

Compilers

- Translates source code (high-level) into low-level
  - Assembly
  - Machine code
  - Bytecode
- Machine-generated machine code
  - Not easy for humans to understand
  - Artifacts of the translation
  - Optmizations
  - Non-intuitive translations (e.g., LEA)

Low-level software

VMs and Bytecodes

- Virtual machines vs. hypervisors
- Between high-level and low-level
- Bytecodes decoded and executed by a program (not CPU)
  - Interpretation
  - Just-in-time (JIT) compilation

Low-level software

VMs and Bytecodes

- Virtual machines vs. hypervisors
- Between high-level and low-level
- Bytecodes decoded and executed by a program (not CPU)
  - Interpretation
  - Just-in-time (JIT) compilation

⊕ Platform independence

⊕ Easier to reverse

⊖ Less intellectual property protection

Low-level software

Operating Systems

- Manages the computer
  - hardware
  - software
- Coordinates elements
- Why such interest in the OS?

Low-level software

Operating Systems

- Manages the computer
  - hardware
  - software
- Coordinates elements
- Why such interest in the OS?
  - Gateway to the outside world

Low-level software

Operating Systems

- Manages the computer
  - hardware
  - software
- Coordinates elements
- Why such interest in the OS?
  - Gateway to the outside world
  - Programming conventions

The Reversing Process

The Reversing Process

- System-level
- Code-level

The Reversing Process

- System-level
  - General structure, areas of interest
  - Running tools on the program
  - Using OS system services
- Code-level
  - Detailed information about a code chunk
  - **Never begin without a narrow focus**
  - Odd combination of art and science
  - Requires mastery of many CS disciplines
  - **Source code is not the end**

The Reversing Process

The Tools

System-Monitoring Tools

Employ OS

sniff, monitor, explore:

- network activity
- file access
- registry
- mutexes
- pipes
- events

Disassemblers

Binary code → mnemonic

Processor-specific

Usually built into debuggers

Debuggers

Common in development

Programs are complex

Many circumstances

Observe execution

Breakpoints, traces

Decompilers

Disassemblers++

Assembly → high-level

Only close to original

Much simply lost

The Reversing Process

Example

```
66 0f 28 d0 f2 0f 59 c9 f2 0f 59 d0 f2 0f 58 d1 f2 0f 51 c2
```

The Reversing Process

Example

```
66 0f 28 d0 f2 0f 59 c9 f2 0f 59 d0 f2 0f 58 d1 f2 0f 51 c2
```

```
66 0f 28 d0   MOVAPD   XMM2,XMM0
f2 0f 59 c9   MULSD    XMM1,XMM1
f2 0f 59 d0   MULSD    XMM2,XMM0
f2 0f 58 d1   ADDSD    XMM2,XMM1
f2 0f 51 c2   SQRTSD   XMM0,XMM2
```

The Reversing Process

Example

```
66 0f 28 d0 f2 0f 59 c9 f2 0f 59 d0 f2 0f 58 d1 f2 0f 51 c2
```

```
 66 0f 28 d0   MOVAPD   XMM2,XMM0
 f2 0f 59 c9   MULSD    XMM1,XMM1
 f2 0f 59 d0   MULSD    XMM2,XMM0
 f2 0f 58 d1   ADDSD    XMM2,XMM1
 f2 0f 51 c2   SQRTSD   XMM0,XMM2
```

```
double func(double a, double b) {
    return sqrt(a*a + b*b);
}
```

The Reversing Process

Example

```
66 0f 28 d0 f2 0f 59 c9 f2 0f 59 d0 f2 0f 58 d1 f2 0f 51 c2
```

```
66 0f 28 d0   MOVAPD   XMM2,XMM0
f2 0f 59 c9   MULSD    XMM1,XMM1
f2 0f 59 d0   MULSD    XMM2,XMM0
f2 0f 58 d1   ADDSD    XMM2,XMM1
f2 0f 51 c2   SQRTSD   XMM0,XMM2
```

```
double func(double a, double b) {
    return sqrt(a*a + b*b);
}
```

$$\sqrt{a^2 + b^2}$$

The Reversing Process

Example

```
66 0f 28 d0 f2 0f 59 c9 f2 0f 59 d0 f2 0f 58 d1 f2 0f 51 c2
```

```
 66 0f 28 d0   MOVAPD   XMM2,XMM0
 f2 0f 59 c9   MULSD    XMM1,XMM1
 f2 0f 59 d0   MULSD    XMM2,XMM0
 f2 0f 58 d1   ADDSD    XMM2,XMM1
 f2 0f 51 c2   SQRTSD   XMM0,XMM2
```

```
double func(double a, double b) {
    return sqrt(a*a + b*b);
}
```

$$\sqrt{a^2 + b^2}$$

"Magnitude of a vector in two dimensions": $\|\langle a, b \rangle\|$

Is Reversing Legal?

Warning

- Always seek legal counsel before beginning a reversing project
- The book only addresses U.S. law
- All labs in class should be legal
  - If you think not, please speak up
- The instructor is not a lawyer

Is Reversing Legal?

Warning

- Always seek legal counsel before beginning a reversing project
- The book only addresses U.S. law
- All labs in class should be legal
  - If you think not, please speak up
- The instructor is not a lawyer

- Attempt to understand impact
  - social
  - economic

Is Reversing Legal?

Cases[4]

1974: Kewanee Oil v. Bicron

“a fair and honest means of starting with the known product and working backwards to divine the process which aided in its development or manufacture.

1989: Bonito Boats v. Thunder Craft

“the competitive reality of reverse engineering may act as a spur to the inventor, creating an incentive to develop inventions that meet the rigorous requirements of patentability.

[4] http://www.taeus.com/articles/software-reverse-engineering-1/

Interoperability

- Difficult even with cooperation
- Requires significant communication among developers

Software platform

Program or hardware device upon which other programs may run
Companies think critically about publishing interface

Interoperability Case Study

Sega v. Accolade

1990: Release of Sega Genesis Console

Sega:

- Chose not to publish interfaces
- Provided only to licensed affiliates
- License required exclusivity (all games!)

Accolade:

- Accolade wished to port and develop for Genesis
- Did not desire to enter an exclusive agreement
- Reverse engineered the console and several cartridges
- Successfully developed and sold several games

Interoperability Case Study

Sega v. Accolade

1991: Sega Sues

Sega argued that intermediate copies from the cartridges was illegal

Ruling

Court favored Accolade

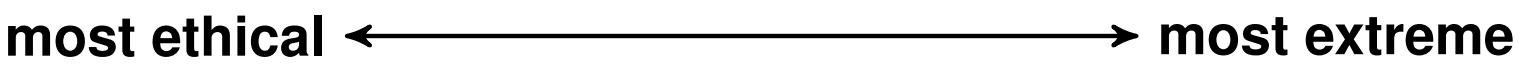
- Accolade's published games included no Sega code
- Favored economic competition from Accolade's work

"the fundamental purpose of the Copyright Act–to encourage the production of original works by protecting the expressive elements of those works while leaving the ideas, facts, and functional concepts in the public domain for others to build on.

Competition

Why should companies invest in innovation if another can reverse and copy it?

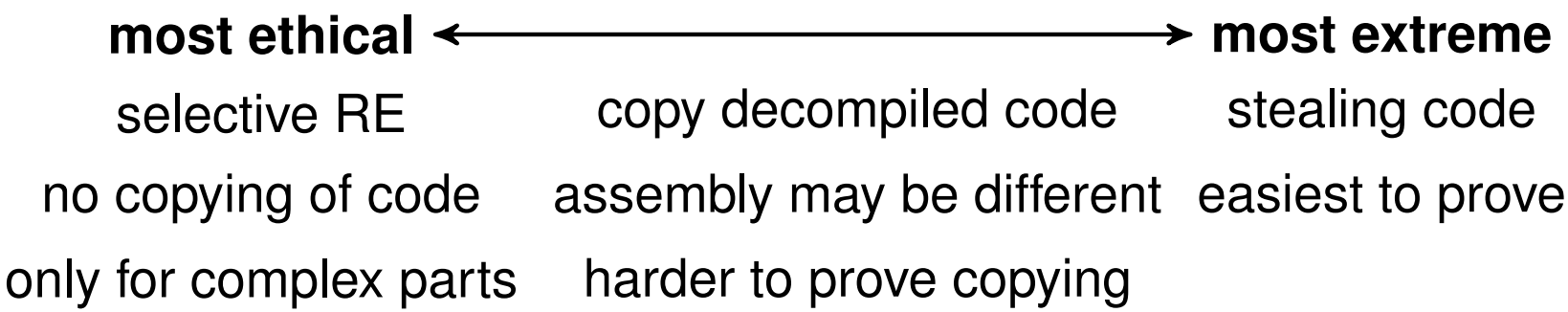
What constitutes reversing for competition?



Competition

Why should companies invest in innovation if another can reverse and copy it?

What constitutes reversing for competition?



Copyright Law

- Designed to protect IP from unauthorized duplication
- Fine line between stealing code and reimplementing it

Interpretation in SRE

- Decompilation is legal (careful your license agreement)
- Opponents typically argue intermediate copies are illegal
  - in-memory and/or on-disk copies
  - argument has not historically held up
    - Sega v. Accolade
    - Sony v. Connectix
  - Intermediates ruled fair use
  - Intermediates also made during installation and execution

Trade Secrets and Patents

Patents

Trade Secrets

Trade Secrets and Patents

Patents

- ⊕ Grants control for up to 20 years
- ⊖ Details must be published
- ⊖ Public domain after expiration

Trade Secrets

- ⊕ Protection from “trade secret misappropriation”
- ⊖ No protection from RE
- ⊖ No protection from independent duplication

Trade Secrets and Patents

Patents

- ⊕ Grants control for up to 20 years
- ⊖ Details must be published
- ⊖ Public domain after expiration

Trade Secrets

- ⊕ Protection from “trade secret misappropriation”
- ⊖ No protection from RE
- ⊖ No protection from independent duplication

- Protections given to any non-patented invention provided sufficient efforts to protect it
- Reverse engineers must obtain the original legitimately from the open market

Digital Millennium Copyright Act (DMCA, 1998)

- Purpose is to protect copyright protection mechanisms
- Closest in U.S. to anti-reverse-engineering law
- Protects mechanisms, not materials or software

Prohibited

- Circumvent copyright protection systems even for personal use
  - Some exceptions
- Develop or make available circumvention technologies
  - Includes keygens

DMCA

Exceptions (1/2)

Interoperability circumvention is permitted for interoperability when it is the only way

Encryption research can circumvent in order to evaluate components

Security testing evaluating or improving security of a computing system

Educational institutions and public libraries evaluation prior to purchase, captioning

DMCA

Exceptions (2/2)

Government investigation DMCA does not prevent agencies from conducting investigations

Regulation circumvention permitted to regulate access to materials by minors

Protection of privacy any protections can be circumvented if the program collects or transmits personal information

DMCA

Cases

2000: Felten v. RIAA

- Hack SDMI challenge: call to test SDMI security by SDMI
- SDMI - DRM for audio, based on watermarks
- Offered $10,000 if ownership of information turned over
- Felten and Princeton chose to forgo prize and published
- SDMI and RIAA filed suit under DMCA
- Team withdrew the paper
- Eventually were allowed to publish at USENIX

DMCA

Cases

2000: Felten v. RIAA

- Hack SDMI challenge: call to test SDMI security by SDMI
- SDMI - DRM for audio, based on watermarks
- Offered $10,000 if ownership of information turned over
- Felten and Princeton chose to forgo prize and published
- SDMI and RIAA filed suit under DMCA
- Team withdrew the paper
- Eventually were allowed to publish at USENIX
- Did SDMI's DMCA suit attempt to reduce security?
- Open publishing vs. closed disclosure?

DMCA

Cases

2001: U.S. v. Sklyarov

- Dmitry Sklyarov worked for ElcomSoft in Moscow
- Reverse engineered Adobe eBook format
- Developed *Advanced eBook Processor* which decrypted it
  - Made it readable by any ordinary PDF reader
  - Essentially removed the protection mechanism
- Adobe filed a complaint
- U.S. sued Skylarov and ElcomSoft

DMCA

Cases

2001: U.S. v. Sklyarov

- Dmitry Sklyarov worked for ElcomSoft in Moscow
- Reverse engineered Adobe eBook format
- Developed *Advanced eBook Processor* which decrypted it
  - Made it readable by any ordinary PDF reader
  - Essentially removed the protection mechanism
- Adobe filed a complaint
- U.S. sued Skylarov and ElcomSoft
- Acquitted by the jury
- (developers didn't know it was illegal)

License Agreements

- License agreements typically prohibit reverse engineering
- Use of the program typically implies acceptance of its terms
- Not clear whether such clauses are enforceable
  - No clear precedent in the U.S.
  - E.U. provides exception for interoperability

License Agreements

Cases

2002: Blizzard v. BnetD[5]

- Tim Jung worked for Internet Gateway, Inc.
- He and two others built BnetD which emulated Battle.net
- BnetD allowed people to play Blizzard games without logging into Battle.net
- Blizzard file complaint - violates DMCA, Blizzard's EULA, and Battle.net's TOU
- EULA prohibited reverse engineering of the product

[5] https://www.eff.org/cases/blizzard-v-bnetd

License Agreements

Cases

2007: Court ruled in favor of Blizzard

- Court ruled click-through agreement was enforceable
- Interoperability exception did not apply since it circumvents protections
- EULA also prohibited use of software with non-Blizzard servers
- BnetD also copied more code than necessary
- Duplicated the play experience - story happenings, etc.

Recap

- Ground rules
- Popular applications
- Typical reversing process
- Tools
- Law

Interface Use Case Study

Oracle v. Google[6]

1995: Release of Java to the public

Sun Microsystems:

- Developed language, Virtual Machine, and APIs
- Standard Edition was dual licensed
- Mobile Edition was commercial licensed
- 2010 Oracle acquired Sun

Android (founded 2003):

- Developed Android OS, Dalvik Virtual Machine
- Developed a translator from Java to Dalvik
- Re-implemented JavaSE APIs for Android/Dalvik
- Avoiding licensing fees
- 2005 Google acquired Android

[6] https://en.wikipedia.org/wiki/Google_LLC_v._Oracle_America,_Inc.

Interface Use Case Study

Oracle v. Google

2010: Oracle Sues

Oracle argued APIs were copyrightable, and claimed patent infringement

First Ruling

Court favored Google

- Jury found non-infringement of all patent claims
- Judge determined APIs were not copyrightable
- The remaining damages were so small, both agreed to $0

“So long as the specific code used to implement a method is different, anyone is free under the Copyright Act to write his or her own code to carry out exactly the same function or specification of any methods used in the Java API. It does not matter that the declaration or method header lines are identical.

Interface Use Case Study

Oracle v. Google

2012: Oracle Appeals

Judge had dismissed API copyrightability despite Jury decision.

- Reversed decision: APIs are copyrightable: “structure, sequence and organization”
- However, a second trial (2016) found that Google’s use was acceptable under fair use

2016: Oracle Appeals Again

“It is undisputed that Google copied verbatim the declaring code of the 37 Java API packages 11,500 lines of Oracle’s copyrighted code.
“The fact that Android is free of charge does not make Google’s use of the Java API packages noncommercial.

Interface Use Case Study

Oracle v. Google

2019: Google Appeals to Supreme Court

Google argued the Appeals Courts had overridden findings of fact, not allowed by Seventh Amendment

- Many companies, academics, professionals, etc., filed briefs supporting Google
- Microsoft argued that finding in favor of Oracle would upend software industry

Final Ruling

Supreme Court favored Google

- Distinguished declaration from implementation