HOW THE BEST HACKERS LEARN THEIR CRAFT

David Brumley

CEO, ForAllSecure
Professor, Carnegie Mellon University
@thedadavidbrumley
George Hotz

- First iPhone Jailbreak
- Playstation 3
- Zero-days in Adobe, Firefox, ...
Richard Zhu

- Mozilla Firefox ('18)
- Microsoft Edge ('17 & '18)
- iOS Safari ('17)
#1 US Team since 2011
#1 Overall 3 of past 7 years
4 DEFCON wins – most wins in DEFCON history
Learning Objectives

1. Understand how top experts use capture the flag competitions for deliberate practice.
2. See how hacking competitions gamify learning computer security.
3. Learn how to set up a system for building a top-ranked hacker culture.
After opening the robot's front panel and looking inside, you discover a small red button behind a tangle of wires. Pressing the button lights up the robot's primary screen. It glows black and quickly flashes blue. A line of small text types out:

```
[ERROR: 0x00000023]
```

The text refreshes and displays the prompt:

```
FILE SYSTEM RECOVERY INITIATED...
FILE SYSTEM COULD NOT BE IDENTIFIED...
PLEASE ENTER FILE SYSTEM FORMAT:
```

Submit!
Basic Knowledge

After opening the robot's front panel and looking inside, you discover a small red button behind a tangle of wires. Pressing the button lights up the robot's primary screen. It glows black and quickly flashes blue. A line of small text types out:

```
[ERROR: 0x00000023]
```

The text refreshes and displays the prompt:

```
FILE SYSTEM RECOVERY INITIATED...
FILE SYSTEM COULD NOT BE IDENTIFIED...
PLEASE ENTER FILE SYSTEM FORMAT:
```

Answer: FAT
Jeopardy-Style CTF

Categories
- Cryptography
  - Caesar Cipher 10 pts
  - Frequency analysis 20 pts
  - RSA Encryption 40 pts
  - RSA Low Exponent 100 pts
- Basic Knowledge
- Reverse engineering
- Network security
- Program Exploitation
- Forensics

Difficulty
Gamify Learning
CTF Principles

1. Applied, deliberate practice
2. Autodidactic learning
3. Creative problem solving
Applied, deliberate practice

Cryptography
Reverse engineering
Network security
Program Exploitation
Forensics
"Buffer Overflow"

User input size programmed
- 10 bytes long

User input given
- 50 bytes long

Class: 90 minutes lecture
1. Sophomore course
2. Students understand concept

Challenge: Apply knowledge
1. Real program buffer size?
2. Create long user input?
3. Create specific attack input?
4. ...
CTF Problem: Show You Can Do It

readasm (615 solves)

At the end of this sequence of instructions, how many bytes separate esp and the stored r in
Assume that we called this function using standard 32-bit x86 calling conventions.

804847c functionname:
804847c: push %ebp
804847d: mov %esp,%ebp
804847f: sub $0x70,%esp
8048482: movl $0x0,0x4(%esp)
804848a: movl $0x8048580,(%esp)

Answer in decimal

HINTS

? You may find this reference informative.
? Put your answer in decimal.
? Not sure about something? Google for it.

Enter flag...
Autodidactic Learning

Auto: self

didactic: learn
Romantic, but not real
Auto-didactic Learning

Richard didn’t know either. He read up.
At the end of this sequence of instructions, how many bytes separate esp and the stored return address on the program's stack? Assume that we called this function using standard 32-bit x86 calling conventions.

804847c functionname:
804847c: push %ebp
804847d: mov %esp,%ebp
804847f: sub $0x70,%esp
8048482: movl $0x0,0x4(%esp)
804848a: movl $0x8048580,(%esp)

Answer in decimal

- HINTS
  - You may find this reference informative.
  - Put your answer in decimal.
  - Not sure about something? Google for it.

Answer: 116
Creative Problem Solving
**Solution vs Result**

Problem in CTFs: “find the flag”
- Solution is flag submitted, like 116 here.
- Wrong! \( \text{flag} \neq \text{solution} \)
- Flag = *result* of the solution

This simplicity is fundamental to creativity
- Check only the results (i.e., the flag)
- Place few constraints on the solution
Creativity in problem solving

\[
2 \times 2 = (\frac{3}{2} + \frac{1}{2})^2
\]

\[
= \frac{3^2}{2^2} + \frac{3}{2} \times \frac{1}{2} + \frac{1}{2^2}
\]

\[
= \frac{9}{4} + \frac{3}{4} + \frac{1}{4}
\]

\[
= 1.75 + 1.5 + 0.25 = 4
\]

All valid approaches
Hack.IM CTF 2012 Example

- Break into PHP-powered website made by organizers
- Reference solution used XPath injection vulnerability

Dutch solution found flaw in PHP, a major programming language
RSA is considered **mathematically** secure.

Software Security

Cryptography

The crypto (math) doesn’t talk about code (app)
Timing attacks against crypto

Suppose my wife asks: “Do I look fat in this outfit”

*Any* hesitation reveals information.
Crypto Software

If key = 1, 1 sec. to decrypt

If key = 2, 2 sec. to decrypt

If key = 3, 3 sec. to decrypt

...
Levels of proficiency

- **Novice**
  - limited experience

- **Awareness**
  - basic knowledge

- **Intermediate**
  - practical application

- **Advanced**
  - applied theory

- **Expert**
  - recognized authority
Jeopardy-Style CTF
Awareness - Intermediate

- Cryptography
- Basic Knowledge
- Reverse engineering
- Network security
- Program Exploitation
- Forensics

Attack-Defense CTF
Advanced - Expert

- Everyone runs same software
- Exploit others = gain points
- Be exploited = loose points
CMU Goals

1. Grow cybersecurity field
2. Identify and attract most promising high school students
3. Systematize the above
<table>
<thead>
<tr>
<th>Level</th>
<th>Challenge</th>
<th>Team Completions</th>
<th>Acquired Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First Contact</td>
<td>1,368</td>
<td>Network Traffic Analysis</td>
</tr>
<tr>
<td>2</td>
<td>CFG to C</td>
<td>1,321</td>
<td>x86 Assembly and Control-Flow Graphs</td>
</tr>
<tr>
<td>2</td>
<td>Try Them All!</td>
<td>1,279</td>
<td>Password Hashes, Salts, and Dictionary Attacks</td>
</tr>
<tr>
<td>3</td>
<td>DDoS Detection</td>
<td>615</td>
<td>Defensive Traffic Analysis</td>
</tr>
<tr>
<td>3</td>
<td>Byte Code</td>
<td>1,146</td>
<td>Program Representation</td>
</tr>
<tr>
<td>3</td>
<td>SQL Injection</td>
<td>571</td>
<td>Command Injection Attacks</td>
</tr>
<tr>
<td>3</td>
<td>RSA</td>
<td>228</td>
<td>RSA Implementation</td>
</tr>
<tr>
<td>4</td>
<td>Overflow 1</td>
<td>216</td>
<td>Buffer Overflow</td>
</tr>
<tr>
<td>4</td>
<td>ROP 1</td>
<td>96</td>
<td>Return-to-libc Attack</td>
</tr>
</tbody>
</table>

Table 2: A summary of PicoCTF challenges and the leaderboard stats.
18,000 High School Students

1. Run PicoCTF.com
2. Top 50 get recommendation
3. Coursework + CTF

Run next picoc tf
System recruits

1. Auto-didactic
2. Demonstrable ability
3. Top talent

Bell-curve of ability
Two Themes

1. CTF problems are a proven, effective way to teach hacking skills

2. You can systematize CTF’s to build your pipeline
Next Actions

• Incorporate CTF’s into your training
Next Actions

• Incorporate CTF’s into your training

• Develop system for identifying talent
  • Build CTF problems representative of skills you care about
  • Use CTF applications to recruit and/or interview
THANK YOU