TEACHING THE PRINCIPLES OF THE HACKER CURRICULUM TO UNDERGRADUATES

Sergey Bratus, Anna Shubina, Michael Locasto
“We posit that the largely undocumented and informal ‘Hacker Curriculum’ contains a number of principles and methods that can prove useful in educating undergraduates in the art of ethical computer security assessment.”
A Positive Definition of “Hacking”

- The skill and ability to question trust and control assumptions expressed in software and hardware systems

- A special skill and mindset
  - Ant farm anecdote

- Not synonymous with criminal behavior
  - Doctor, policeman, locksmith
Imperative to learn from this community rather than shun it.

This can be done in a principled fashion that differs from traditional CS pedagogy and that will not turn students into criminals.
Providing a “Security Culture Shock”

- CS and IT programs have become efficient at imparting marketable skills
  - Under time pressure to cover a rapidly expanding field of knowledge
  - Knowledge is compartmentalized
  - Teaching environments tend to hide the messy details and engineering complexity
  - No time to “waste” on puzzling out failures “unrelated” to main task
  - Students are taught success cases & paths (and possibly some anticipated failure states)
This teaching approach conditions students to trust their computing environment

- Even if it masks complexity to reduce confusion
- Students naturally trust API boundaries and this de facto causes them to become boundaries of competence and expertise

- *Nothing is more damaging to a security mindset than this type of conditioning*
Human operator

- main()
- libc_func()
- system_call()
- VFS / sys_func()
- dev driver code

Actual system state

User-level rootkits
DLL hijacking
Simple kernel rootkits
Advanced kernel rootkits

Humans operator
Three Examples

- The SISMAT program
- Dartmouth’s CSI “student red team” program
- Our colleagues’ experience bringing students to hacker conferences
The SISMAT Program

- An intensive undergraduate cyber-security program involving education, practical experience, and research

- We teach a variety of topics
  - Hacking tools, program analysis, PKI, OS kernel code, x86 assembly, network protocols and traffic analysis, security management
Example SISMAT Exercises

- SQL/HTML Injection
- OpenSSL “roll-your-own-CA”
- Forging and digitally signing email
- Detailed analysis of a real, extensive intrusion
- Authorization policy in Unix
- Debugging and analysis of libpng and nullhttpd vulnerabilities (gdb, working shellcode, etc.)
Dartmouth CSI

- Group of Dartmouth undergraduates
  - Supervised and mentored by PKCS personnel, CS Dept researchers and staff
  - Students sign an NDA and undergo a background check
  - Actual red-team style security assessments of existing and proposed information infrastructure
The “Hacker Curriculum”

- A largely undocumented set of principles and methods for learning about information security

- Not an ad-hoc collection of hacks, but rather a complex, rich (but often implicit) structure

- Approaches traditional CS/CE/CIS topics from a variety of different perspectives
Focus: Two Hacker Curriculum Principles

- “learning from failure modes”
  - Teaching students how systems work under extraordinary conditions can be more revealing than simply highlighting how the system works under perfect conditions

- “violate API boundaries”
  - These become boundaries of expertise and core competence
Rogue clients
Raw sockets
Frame injection

IDS/IPS

Attacker

Target

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data link
1. Physical
Why Is This Different than CS?

- Do we need yet another curriculum?
- Do undergraduates really need to learn this stuff?
- Are we ethical responsible or legally culpable if they abuse this knowledge?
- Undergraduates don’t have the ethical grounding not to resist the Dark Side
- This isn’t science, just engineering
- CS students should be concerned with learning science, algorithms, and principles, not tools
Contrast Approach to Networks

Hacker Approach

- Reconnaissance
  - Hosts, services, filtering rules
- Discovery
  - Adverse conditions, misconfigurations, router ACLs, firewalls, configuration

CS Approach

- The OSI model
- Protocol Diagrams
- Packet Header Figures
- Example Socket Code
Networking Through Failure Modes

Gentle introductions to network communication almost wholly concentrate on the way things *should work* rather than the ways that they might be *made to fail*.

The hacker approach is a mission of discovery of the actual model in use rather than a clean, reliable model drawn from textbook pages.
Networking Through Cross-Layer

Students tend to learn about different layers of the network in different courses (e.g., OS networking, Unix userland networking, application-level protocols (Web/HTTP/SOAP))

The hacker approach takes a cross-layer analysis approach to understanding the data structures involved in transporting data up and down the stack.
Cross-Layer Networking Observations

- Crafting and injecting arbitrary and malformed packets into the network
  - Gain understanding of code, systems, and toolchains that manipulate network data
  - Important from a culture-shock perspective
  - Students realize that network packets and frames are malleable and that networks have almost no trust guarantees
tcpdump
wireshark

ettercap
dsniff/
dnsspoof

fragrouter

libpcap

packet
forwarding

libnet

raw sockets

packet
capture

sniffing/
interception

IP
forwarding/
routing

injection/
spoofing
Key idea: achieve visibility of network packets and state as a variety of failed connections occur

- Observe results of connections to non-existent hosts inside and outside LAN, with missing services (e.g., DNS, ARP)
- Students learn basic scanning and firewalking techniques
- Start to look for and understand failure events with confidence rather than dismissing them as unintelligible

What happens to shortest path or routing algorithms when nodes lie about their neighbors?
Contact

mlocasto@gmu.edu
sergey@cs.dartmouth.edu
ashubina@cs.dartmouth.edu

http://www.hackercurriculum.org
http://www.ists.dartmouth.edu/events/sismat/