Security

CS 144 Section, Fall 2010
pthreads Reminders

- pthreads = POSIX threads API
- You’ll need pthreads for lab 5, but most of the details are done for you in the example code
- Remember to acquire and release locks when accessing the NAT table
- Suggestion: use coarse locking (a single lock is fine)
- See given sr_nat.c and sr_nat.h for details
Today: Security topics

- Zombie botnets
- JavaScript Cross-site Request Forgery
IRC

- IRC: Internet Relay Chat [RFC 1459]
- Created in 1988 by Jarkko Oikarinen
- Still used today, especially by open source community
- Very simple ASCII protocol (usable with telnet)
- Thus, it’s easy to write IRC clients
- Telnet demo
Zombie Botnets

- Zombies are malicious drone programs running on unsuspecting host computers
- Typically connect to an IRC server and wait for commands
- Can be installed a number of ways: malware, network attacker injecting commands, or exploiting a vulnerability
- **Example vulnerability** in phpBB from 2004
Control Commands

- `mail` – send an email
- `download` – download a file
- `exec` – run a command
- `udpflood` – send UDP packets at a target
- `pscan` – do a port scan on a target
- Just as good as having shell access to the machine!
Zombie IRC Log

--- Day changed Sat Jun 12 2010
17:58 -!- krobelus [~via@187.126.192.104] has joined #botcrew
17:59 -!- [Y]inurl6398 [~inurl3908@189.126.193.10] has joined #botcrew
17:59 -!- [Y]inurl2290 [~inurl1312@hm2208.locaweb.com.br] has joined #botcrew
17:59 -!- [Y]inurl9724 [~inurl6074@hm231.locaweb.com] has joined #botcrew
18:00 < krobelus> .user 190985
18:00 < krobelus> .udpflood 189.1.164.34 900 900 1500
18:00 < [Y]inurl12290> [UdpFlood Started!]
18:00 < [Y]inurl16398> [UdpFlood Started!]
18:00 < [Y]inurl19724> [UdpFlood Started!]

- **Notice zombie hosts:** locaweb.com.br
- **whois** 189.1.164.34
- **Consider this scenario when configuring firewall policies**
JavaScript Cross-site Request Forgery (CSRF)

- Can be done without violating the Same Origin Policy
CSRF Demo
Protecting Against CSRF

• Include a special hidden token that is submitted with the form

• Usually a hash of some user-specific data and a secret key

<input type='hidden' autocomplete='off'
name='post_form_id'
value='3549e334daee0ef6dbc772c45cf517bf' />
Summary

• Security concerns exist at all layers in the networking stack

• “Secure” protocols are sometimes not enough: phpBB attack works over SSL, too

• Network attackers can see and modify network traffic and cause unexpected behavior, including both of the examples shown here