Distributed Denial of Service Attacks
CS 239
Security for Networks and System Software
May 15, 2002

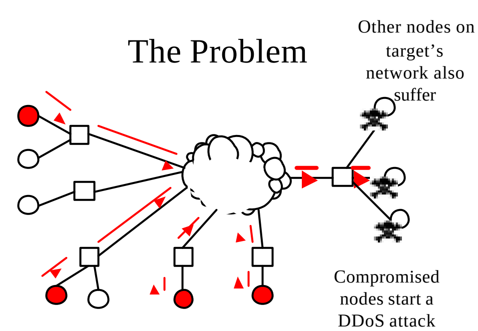
Outline

- Introduction
- Characteristics of DDoS attacks
- Some examples
- Proposed prevention methods

Introduction

- DDoS is a relatively new kind of attack
  – First seen at small scale late in 99
- Use standard denial of service tools
  – SYN floods, smurf attacks, etc.
- Combined with not-very-sophisticated distributed systems technology
- Resulting in an extremely effective attack

The Problem



Other Elements of Such Attacks

- Each attacking machine can spoof its IP address
- Typically under control of a single master machine
  – Why is this "better" than launching from the attacker's own machine?
- Often able to use different kinds of attacks

Why Are Distributed Denial of Service Attacks Hard to Handle?

- Single machine denial of service attacks are hard to handle
- Spoofed IP addresses makes it harder
- The Internet offers few or no tracing tools
- Hacker toolkits make it trivial to compromise many machines

Sample Distributed Denial of Service Toolkits

- Trinoo
- Tribe Flood Network
- Stacheldraht

Trinoo

- An early example
- Relatively unsophisticated
- But still effective
- Doesn't spoof IP addresses
- Uses UDP flooding attacks
  - Basically, sending streams of UDP packets at random ports

Trinoo Masters and Daemons

- The machines actually sending the UDP packets are daemons
- The daemons are controlled by one or more masters
- Master machines start and stop attacks
  - And specify the victim
- Daemons store encrypted list of acceptable masters

Tribe Flood Network (TFN)

- Somewhat more sophisticated than trinoo
- Also uses master and daemon concept
- But can spoof IP addresses
- And can exploit several different weaknesses
  - TCP SYN flood, ICMP echo request flood, smurf attacks, plus UDP floods
- Master/daemon communications sometimes encrypted

Stacheldraht

- German for barbed wire
- Derived, apparently, from Tribe Flood Network
- Added encryption to master/daemon communications before TFN did
- Uses similar attacks to TFN

Where Did the Toolkits Come From?

- A German hacker who calls himself Mixter wrote at least some of them
  - TFN, at least
- Other hackers altered his code or wrote their own
- After authors fiddled around a bit, they posted the kits to hacking sites

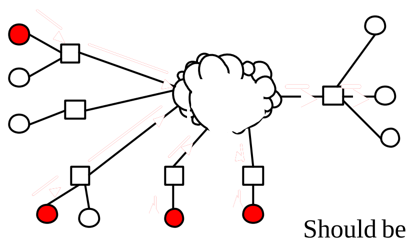
Effects of Distributed Denial of Service Attacks

- Successfully launched against Yahoo, CNN, ETrade, many other sites
- Less successfully launched against Microsoft
  - Attacker didn't have enough client machines
- Attacks occur regularly

Combating Distributed Denial of Service Attacks

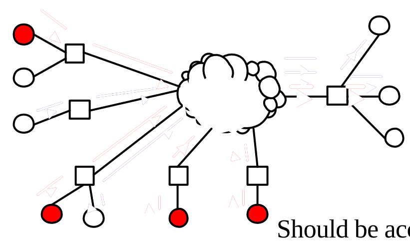
- Desirable properties of solution
- Approaches

Desirable Solution Properties



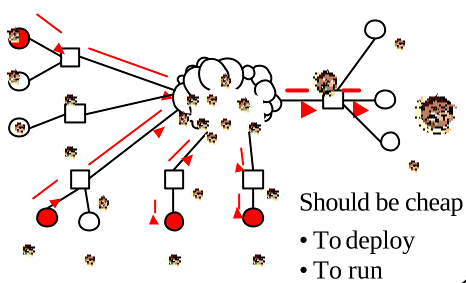
Should be quick

Desirable Solution Properties



Should be accurate

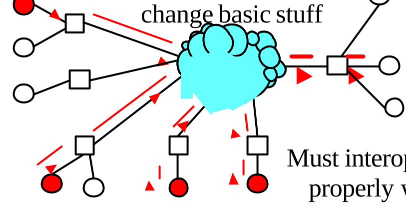
Desirable Solution Properties



Should be cheap
- To deploy
- To run

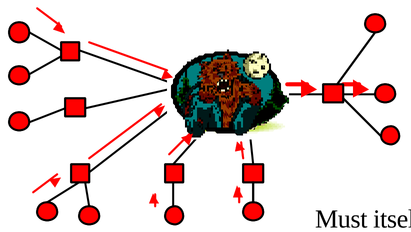
Desirable Solution Properties

Not realistic to change basic stuff



Must interoperate properly with existing Internet technology

Desirable Solution Properties



Must itself be secure

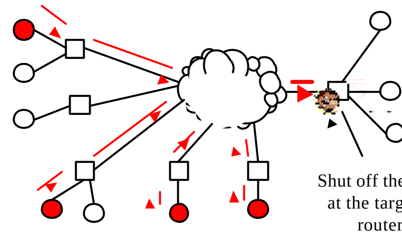
Candidate Approaches

- Filtering at the target
- Tracing approaches
- Pushback approaches
- Filtering near source
- Cooperative approaches
- Public hygiene approaches
- Law enforcement approaches

Filtering at the Target

- When attack is detected, filter it
- How?
  - Based on source IP addresses
  - Based on other header information
  - Based on packet payload information
- Modern routers can do this filtering

Filtering Solutions



Shut off the flow
at the target's
router

Problems With Filtering Solutions

- Can only be reactive
- Often requires assistance of third party
  - ISP provider or backbone site
- Can't filter everything always
- More clever attacks could bypass any simple filter

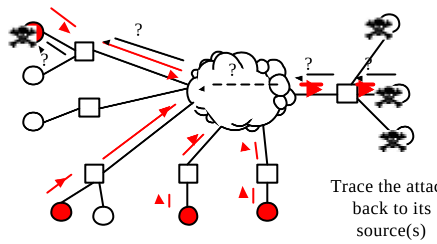
Tracing Approaches

- Find the sending sources and shut them down
- Requires tracing the attack packets back through the network
- Not simple with today's technology
- Smart attackers only attack for a while
  - Leaving nothing to trace

Basics of Tracing

- Identify an attack packet
- Check its IP address
  - If not forged, take external action
  - But it's probably forged
- Ask next upstream router where it came from
- And that router must ask the previous router

Tracing Solutions



Trace the attack back to its source(s)

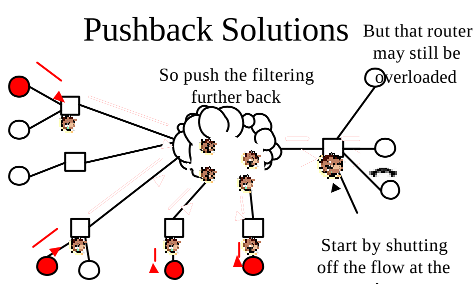
Problems With Tracing Solutions

- No automated tools to do this
- "Asking a router" amounts to a phone call to a system administrator
- Ultimately requires help of backbone providers
- In wide DDOS, may have to trace hundreds of attack streams

Pushback Approaches

- Install filtering at router close to target
- That router asks upstream routers to install filters
  - Which relieves the burden on target's router
- Filters can be pushed further back, as needed
- Can rate limit, rather than filter

Pushback Solutions



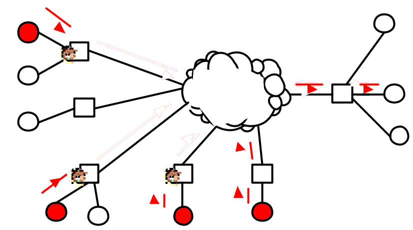
Problems With Pushback Approaches

- Requires cooperation among parties who normally don't cooperate
- Must address security flaws
- Like other types of filtering, may filter the wrong stuff
  - And, with this approach, may get a lot of it

Filtering Near the Sources

- Try to detect the problem close to the sites that are creating the traffic
- Rate limit at routers close to the problem sites
- A distributed solution to a distributed problem
- Routers near attackers may have better information

Source Side Filtering Solutions



Shut off the flow at multiple entry points

Problems With Filtering Near the Sources

- Requires deployment at many sites to be effective
- Trying to detect the problem far away from where it occurs
- Might be foolable from outside the local network
  - Turning the defense tool into an attack tool

Cooperative Approaches

- Gather information from many different sources
- Analyze the total information to understand what's going on
- Apply some subset of previous mechanisms to solve the problem

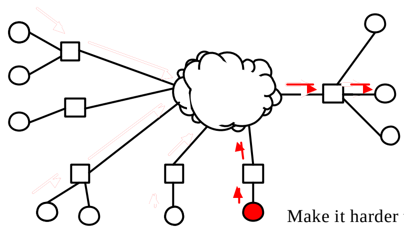
Problems With Cooperative Approaches

- Must leverage off other approaches
  - Possibly inheriting their problems
- Some information provided may be untrustworthy
- Presumes some network connectivity
  - Will that be available during an attack?

Public Hygiene Approaches

- A longer-term solution
- Make sure that it's harder to launch attacks
- Make sure it's harder to spoof IP addresses
- Basically, make sure everyone on the Internet has secure machines

Public Hygiene Solutions



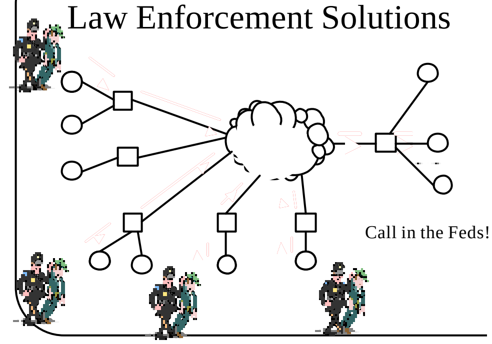
Problems With Public Hygiene Approaches

- Only work well if a high percentage of all sites follow them
- Only work as long as no new vulnerabilities are discovered
- Some of the prophylactic measures are limiting to those who apply them
  - And they're not directly getting the benefits

Law Enforcement Approaches

- Call in the FBI
- Have them trace down the culprit and toss him in jail
- That'll teach him!

Law Enforcement Solutions



Problems With Law Enforcement Approaches

- The law, in its majesty, moves slowly
  - Even by human standards
- This kind of investigation is inherently costly
  - And thus can't often be done
- Smart attackers may be very, very hard to find

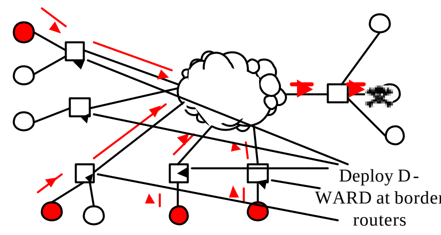
A Sample Approach

- D-WARD
- Being developed here at UCLA
- One of the family of approaches that works close to sources

Basic Ideas Behind D-WARD

- Deploy at routers at exit points of networks
- Observe two-way traffic to particular destinations
- If “bad” traffic patterns, apply rate limits
- Observe how “bad” traffic behaves when limited
  - If well-behaved, relax limit
  - If poorly behaved, set higher limit

D-WARD in our Example



Detecting Problems

- D-WARD observes all traffic through router
  - Since border router, volume is usually reasonable
- Track traffic by destination address
  - Which won’t be forged, unlike source
- Over time, compare pattern of traffic to known good patterns

What Is Good Traffic?

- For TCP, a small ratio of packets sent to packets received
  - Due to ACKS
- For things like ICMP, similar
- But what about UDP?
  - A challenging problem for the research

What Does D-WARD Do When It Finds a Problem?

- Apply a rate limit to all traffic flowing towards destination address
  - Set sufficiently low to limit problems at possible target
  - But some traffic still flows
- Basic idea gives “fair share” to all offered traffic
  - Which would cause attack traffic to push out good traffic

Giving Preferential Treatment to Good Traffic

- Could observe flows to target on a source IP address level
  - Keep separate counts for each source IP address observed
- What will happen if we do that?
- Are there some problems with realistic routers here?

What Happens Next?

- D-WARD observes the local network's response to the rate limit
  - Well-behaved flows back off when rate limits are applied
  - Does this flow?
- Gradually ease rate limit if the traffic is well-behaved
- Keep it or increase it if poorly behaved

Status of System

- Prototype built
  - In Linux router
- Experiments have been performed
- Works quite well
  - Able to shut down large percentage of all attack traffic
- Good flows from other places get through
  - Even if their packets are indistinguishable from attack packets

Challenges for D-WARD

- Differentiation and preferential treatment for good flows
- Deployment
- Security issues