Denial-of-Service (DoS)

CS 161 - Computer Security

Profs. Vern Paxson & David Wagner

TAs: John Bethencourt, Erika Chin, Matthew Finifter, Cynthia Sturton, Joel Weinberger

http://inst.eecs.berkeley.edu/~cs161/

Feb 22, 2010

Announcements

- Section 108 (Tu 2-3PM, TA: Joel) is being moved from 70 Evans to 122 Barrows for the **next three weeks**
  - Will go back to 70 Evans on March 16

The Threat of Denial-of-Service

- Denial-of-Service (DoS, or "*doss*"): *keeping someone from using a computing service*
- Two basic approaches available to an attacker:
  - Deny service based on a program flaw
    - E.g., supply an input that crashes a server
  - Deny service based on resource exhaustion
    - E.g., consume CPU, memory, disk, network
- How broad is this sort of threat?
  - *Very*: **huge** attack surface
- We do though need to consider our threat model …
  - What might motivate a DoS attack?

Motivations for DoS

- Showing off / entertainment / ego
- Competitive advantage
  - Maybe commercial, maybe just to win
- Vendetta / denial-of-money
- Extortion
- Political statements
- Impair defenses
- Warfare

DoS Defense in General Terms

- Defending resources from exhaustion can be **really** hard. Requires:
  - *Isolation mechanisms*
  - *Reliable identification* of different users
- Need to beware of asymmetries, where attackers can consume victim resources with little comparable effort
  - Makes DoS easier to launch
- One dangerous form of asymmetry: **amplification**
  - Attacker leverages system's structure to pump up the load they induce on a resource

DoS & Operating Systems

- How could you DoS a multi-user Unix system on which you have a login?
  - `# rm -rf /`
    - (if you have root - but then just “halt” works well!)
  - ```
    char buf[1024];
    int f = open("/tmp/junk");
    while (1) write(f, buf, sizeof(buf));
    ```
    - Gobble up all the disk space!
  - `while (1) fork();`
    - Create a zillion processes!
  - Create zillions of files, keep opening, reading, writing, deleting
    - Thrash the disk
  - … doubtless many more
- Defenses?
  - Isolate users / impose quotas
    ```

DoS & Networks

- How could you DoS a target's Internet access?
  - Send a zillion packets at them
  - Internet lacks isolation between traffic of different users!
- What resources does attacker need to pull this off?
  - At least as much sending capacity ("bandwidth") as the bottleneck link of the target's Internet connection
    - Attacker sends maximum-sized packets
  - **Or**: overwhelm the rate at which the bottleneck router can process packets
    - Attacker sends minimum-sized packets! (in order to maximize the packet arrival rate)

Defending Against Network DoS

- Suppose an attacker has access to a beefy system with high-speed Internet access (a “big pipe”). They pump out packets towards the target at a very high rate.
- What might the target do to defend against the onslaught?
  - Install a network filter to discard any packets that arrive with attacker’s IP address as their source
    - Or it can leverage *any other pattern* in the flooding traffic that’s not in benign traffic
  - Filter = *isolation mechanism*
  - Attacker’s IP address = means of *identifying* misbehaving user

Filtering Sounds Pretty Easy …

- … but it's not. What steps can the attacker take to defeat the filtering?
  - Make traffic appear as though it's from many hosts
    - Spoof the source address so it can't be used to filter
      - Just pick a random 32-bit number of each packet sent
    - How does a defender filter this?
      - They don't!
      - Best they can hope for is that operators around the world implement anti-spoofing mechanisms (today about 75% do)
  - Use many hosts to send traffic rather than just one
    - Distributed Denial-of-Service = **DDoS** ("dee-doss")
    - Requires defender to install complex filters
    - How many hosts is "enough" for the attacker?
      - Today they are very cheap to acquire … :-(

Botnets Beat Spartan Laser on *Halo 3*

By Kevin Poulsen ✉ February 4, 2009 | 12:13 pm | Categories: Cybarmageddon!

What's the most powerful weapon you can wield when playing *Halo 3* online?

I know. You can control the entire map with a battle rifle and a couple of sticky grenades. But that teeny-bopper you just pwned has you beat with the tiny botnet he leased with his allowance money.

Extortion via DDoS on the rise

By Denise Pappalardo and Ellen Messmer, Network World, 05/16/05

Criminals are increasingly targeting corporations with distributed denial-of-service attacks designed not to disrupt business networks but to extort thousands of dollars from the companies.

Ivan Maksakov, Alexander Petrov and Denis Stepanov were accused of receiving $4 million from firms that they threatened with cyberattacks.

The trio concentrated on U.K. Internet gambling sites, according to the prosecution. One bookmaker, which refused to pay a demand for $10,000, was attacked and brought offline--which reportedly cost it more than $200,000 a day in lost business.

NOV 06

8

DDoS makes a phishing e-mail look real

Posted by Munir Kotadia @ 12:00

0 comments

Just as Internet users learn that clicking on a link in an e-mail purporting to come from their bank is a bad idea, phishers seem to be developing a new tactic -- launch a DDoS attack on the Web site of the company whose customers they are targeting and then send e-mails "explaining" the outage and offering an "alternative" URL.

November 17th, 2008

Anti fraud site hit by a DDoS attack

Posted by Dancho Danchev @ 4:01 pm

Categories: Botnets, Denial of Service (DoS), Hackers, Malware, Pen testing...

Tags: Security, Cybercrime, DDoS, Fraud, Bobbear...

The popular British anti-fraud site **Bobbear.co.uk** is currently under a DDoS attack (distributed denial of service attack) , originally launched last Wednesday, and is continuing to hit the site with 3/4 million hits daily from hundreds of thousands of malware infected hosts mostly based in Asia and Eastern Europe, according to the site's owner. Targeted DDoS attacks against anti-fraud and volunteer cybercrime fighting communities clearly indicate the impact these communities have on the revenue stream of scammers, and with Bobbear attracting such a high profile underground attention, the site is indeed doing a very good job.

Amplification: Network DoS

- One technique for magnifying flood traffic: leverage Internet's *broadcast functionality*

```
cory 1 % ping -s 128.32.48.0
PING 128.32.48.0: 56 data bytes
```

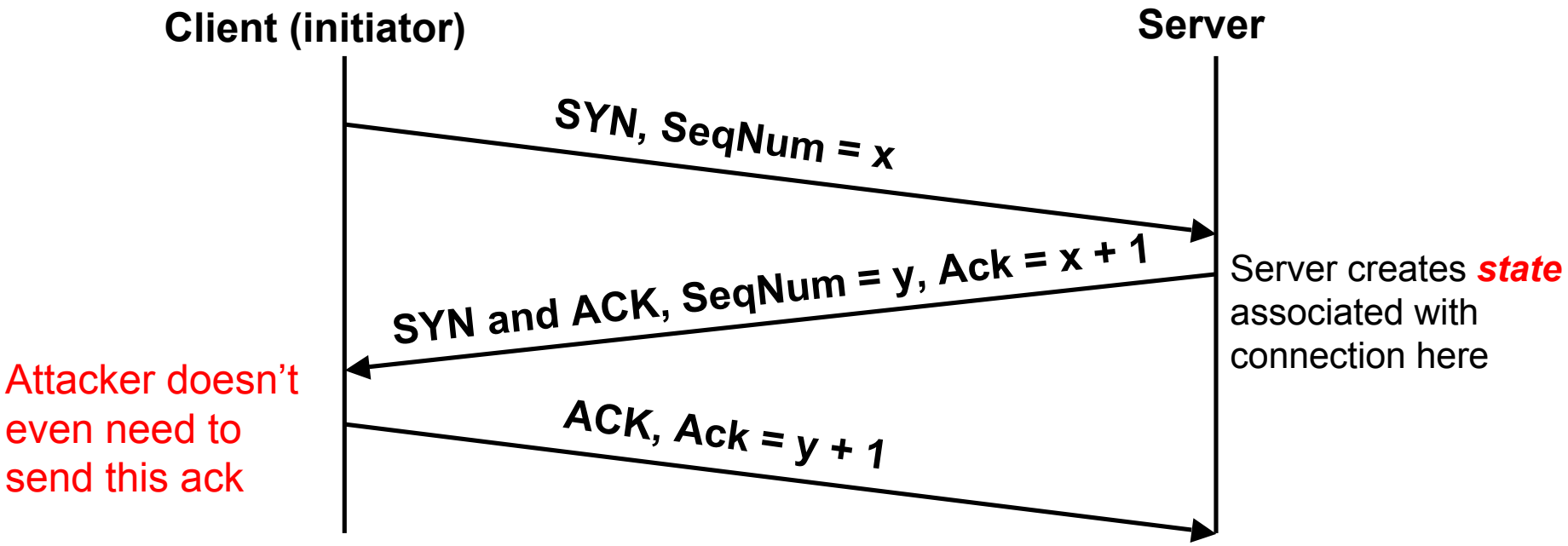
```
cory 1 % ping -s 128.32.48.0
PING 128.32.48.0: 56 data bytes
64 bytes from cory.EECS.Berkeley.EDU (128.32.48.187): icmp_seq=0. time=0.599 ms
64 bytes from verify.EECS.Berkeley.EDU (128.32.48.124): icmp_seq=0. time=1.66 ms
64 bytes from claude.EECS.Berkeley.EDU (128.32.48.242): icmp_seq=0. time=3.50 ms
64 bytes from wiener.EECS.Berkeley.EDU (128.32.48.173): icmp_seq=0. time=4.89 ms
64 bytes from cronus-48.CS.Berkeley.EDU (128.32.48.21): icmp_seq=0. time=6.24 ms
64 bytes from skyros.EECS.Berkeley.EDU (128.32.48.189): icmp_seq=0. time=7.60 ms
64 bytes from citrissrv4.EECS.Berkeley.EDU (128.32.48.138): icmp_seq=0. time=8.95 ms
64 bytes from kea.EECS.Berkeley.EDU (128.32.48.161): icmp_seq=0. time=10.3 ms
64 bytes from rhea-48.CS.Berkeley.EDU (128.32.48.23): icmp_seq=0. time=11.7 ms
64 bytes from mercury2.EECS.Berkeley.EDU (128.32.48.116): icmp_seq=0. time=13.1 ms
64 bytes from transacct.EECS.Berkeley.EDU (128.32.48.243): icmp_seq=0. time=14.4 ms
64 bytes from erso-stag.EECS.Berkeley.EDU (128.32.48.235): icmp_seq=0. time=15.8 ms
64 bytes from pems-pl.EECS.Berkeley.EDU (128.32.48.206): icmp_seq=0. time=17.1 ms
64 bytes from pemsdc.EECS.Berkeley.EDU (128.32.48.199): icmp_seq=0. time=18.4 ms
64 bytes from pemscs.EECS.Berkeley.EDU (128.32.48.156): icmp_seq=0. time=19.8 ms
64 bytes from erso-dev.EECS.Berkeley.EDU (128.32.48.188): icmp_seq=0. time=21.1 ms
64 bytes from kynthos.EECS.Berkeley.EDU (128.32.48.125): icmp_seq=0. time=22.6 ms
64 bytes from pemsdb.EECS.Berkeley.EDU (128.32.48.157): icmp_seq=0. time=24.1 ms
64 bytes from ildap2.EECS.Berkeley.EDU (128.32.48.164): icmp_seq=0. time=25.5 ms
64 bytes from pulsar.EECS.Berkeley.EDU (128.32.48.149): icmp_seq=0. time=26.8 ms
64 bytes from quasar.EECS.Berkeley.EDU (128.32.48.145): icmp_seq=0. time=28.2 ms
64 bytes from c199.EECS.Berkeley.EDU (128.32.48.169): icmp_seq=0. time=29.6 ms
64 bytes from boron.EECS.Berkeley.EDU (128.32.48.118): icmp_seq=0. time=31.0 ms
64 bytes from silicon2.EECS.Berkeley.EDU (128.32.48.204): icmp_seq=0. time=32.4 ms
64 bytes from print199md-cc.EECS.Berkeley.EDU (128.32.48.196): icmp_seq=0. time=33.8 ms
64 bytes from silicon.EECS.Berkeley.EDU (128.32.48.237): icmp_seq=0. time=35.2 ms
64 bytes from print197m.EECS.Berkeley.EDU (128.32.48.227): icmp_seq=0. time=36.6 ms
64 bytes from print144ma.EECS.Berkeley.EDU (128.32.48.228): icmp_seq=0. time=38.0 ms
64 bytes from cory115-1-gw.EECS.Berkeley.EDU (128.32.48.1): icmp_seq=0. time=39.4 ms
64 bytes from print199ma.EECS.Berkeley.EDU (128.32.48.201): icmp_seq=0. time=40.8 ms
64 bytes from print199mb.EECS.Berkeley.EDU (128.32.48.202): icmp_seq=0. time=42.2 ms
64 bytes from print199md.EECS.Berkeley.EDU (128.32.48.213): icmp_seq=0. time=43.6 ms
64 bytes from mshop-print.EECS.Berkeley.EDU (128.32.48.219): icmp_seq=0. time=44.9 ms
```

Amplification: Network DoS

- One technique for magnifying flood traffic: leverage Internet's *broadcast functionality*
- How does an attacker exploit this?
  - Send traffic to the broadcast address and spoof it *as though the DoS victim sent it*
  - All of the replies then go to the victim rather than the attacker's machine
  - Each attacker pkt yields dozens of flooding pkts
- Another example: DNS lookups
  - *Reply is often much bigger than request*
  - So attacker spoofs request seemingly from the target
    - Small attacker packet yields large flooding packet

Transport-Level Denial-of-Service

- Recall TCP's 3-way connection establishment handshake
  - Goal: agree on initial sequence numbers
- So a single SYN from an attacker suffices to force the server to spend some memory



TCP *SYN Flooding*

- Attacker targets *memory* rather than network capacity
- Every (unique) SYN attacker sends burdens the target
- What should target do when it has no more memory for a new connection?
- No good answer!
  - Refuse new connection? Legit new users can't access service
  - Evict old connections to make room? Legit old users get kicked off

TCP SYN Flooding, con't

- How can the target defend itself?
- Approach #1: make sure they have **tons of memory**!
  - How much is enough? Depends on resources attacker can bring to bear
- Approach #2: identify bad actors & refuse their connections
  - Hard because only way to identify them is based on IP address
    - We can't for example require them to send a password because doing so requires we have an established connection!
  - For a public Internet service, who knows which addresses customers might come from?
  - Plus: attacker can spoof addresses since they don't need to complete TCP 3-way handshake
- (Approach #3: don't keep state! We'll see such a technique later in the course, "*SYN cookies*")

Application-Layer DoS

- Rather than exhausting network or memory resources, attacker can overwhelm a service's processing capacity
- There are many ways to do so, often at little expense to attacker compared to target (asymmetry)

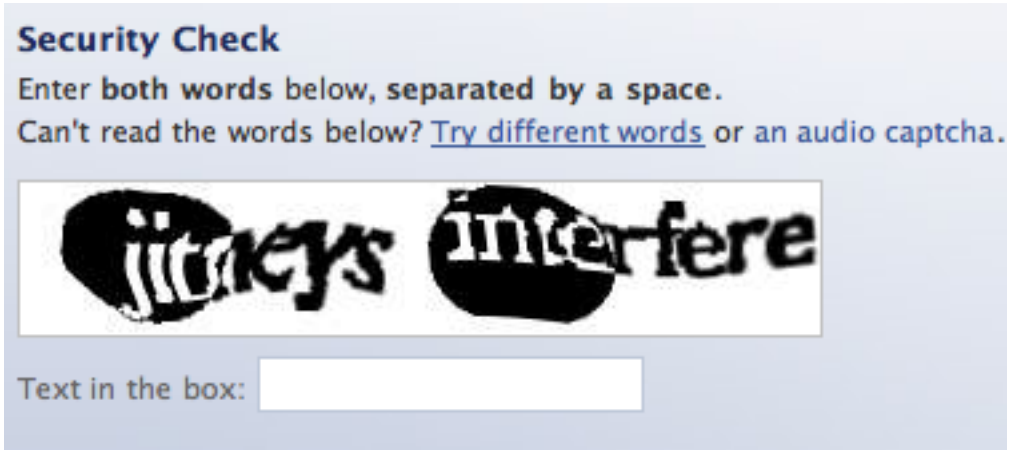

reddit
hot
new
browse
stats
This link runs a slooow SQL query on the RIAA's server. Don't click it; that would be wrong. (tinyurl.com)
814 points posted 8 days ago by keyboard_user 211 comments

Application-Layer DoS

- Rather than exhausting network or memory resources, attacker can overwhelm a service's processing capacity
- There are many ways to do so, often at little expense to attacker compared to target (asymmetry)
- Defenses against such attacks?
- Approach #1: Only let legit users to issue expensive requests
  - Relies on being able to identify/authenticate them
  - Note: that *this itself might be expensive*!
- Approach #2: At least require request to come from a human rather than a program ("bot")

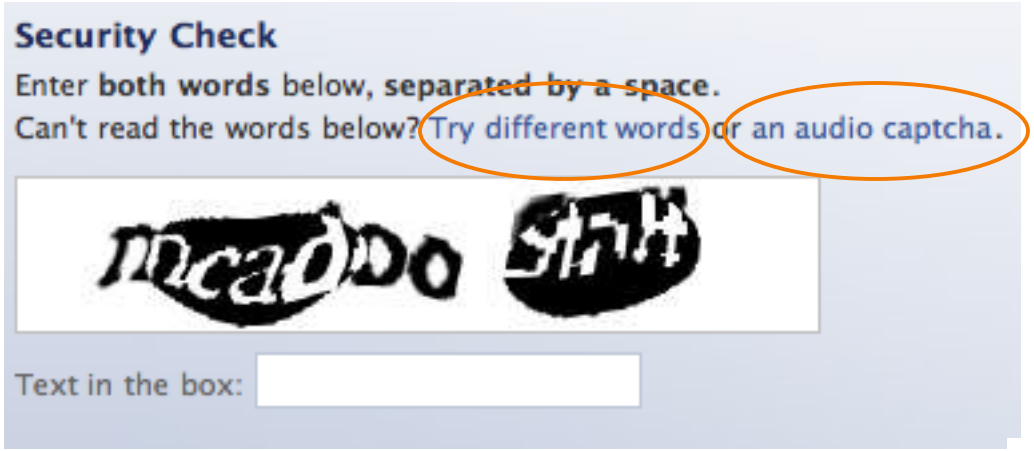
CAPTCHAs

- *Reverse Turing Test*: present "user" a challenge that's easy for a human to solve, hard for a program to solve
- One common approach: distorted text that's difficult for character-recognition algorithms to decipher



Issues with CAPTCHAs

- Inevitable arms race: as solving algorithms get better, defense erodes, or gets harder for humans



  - *Accessibility*: not all humans can see!
  - *Granularity*: not all bots are bad! (e.g., crawlers)

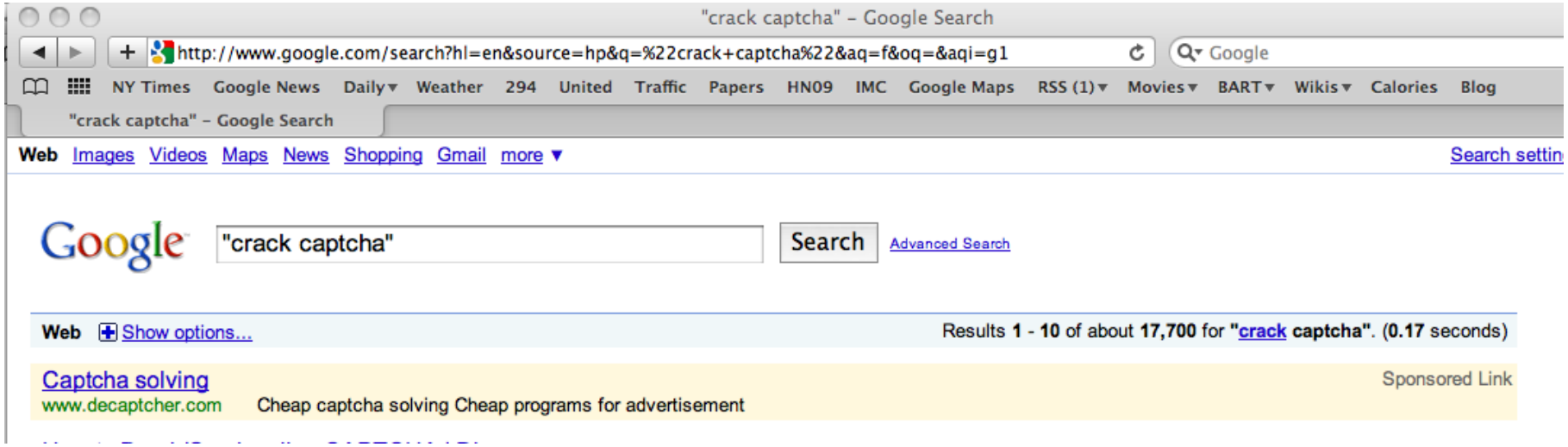
Issues with CAPTCHAs, con't

- If generating a CAPTCHA is somewhat expensive, the mechanism itself is a DoS vulnerability!


reddit
hot
new
browse
stats
Clicking this link loads 120,000 copies of the RIAA's captcha. Clicking would be wrong, don't do it. (antisocial.propagation.net)
452 points posted 4 days ago by mridlen 292 comments

Issues with CAPTCHAs, con't

- If generating a CAPTCHA is somewhat expensive, the mechanism itself is a DoS vulnerability!
- In general, any security mechanism that takes significant resources (CPU or state in memory) can itself introduce a DoS vulnerability
- Final problem: CAPTCHAs are inherently vulnerable to *outsourcing* attacks
  - Attacker gets real humans to solve them



Using the advertisement in blogs, social networks, etc significantly increases the efficiency of the business. Many services use pictures called CAPTCHAs in order to prevent automated use of these services.

Solve CAPTCHAs with the help of this portal, increase your business efficiency now!

Follow these steps:

Register
Login and follow the link inside to load funds to your account.
Your request will be processed ASAP.

You pay for correctly recognized CAPTCHAs only
The price is $2 for 1000 CAPTCHAs. We accept payments from $10.

If you use a third-party software the price could be different, contact the software vendor for more information.

Hi! I want to bypass captcha from my bots. Bots have different IPs. Is it possible to use your service from many IPs?

> We have no restrictions about IP: with DeCaptcher you can bypass CAPTCHA from as many IPs as you need.

Hi. I need to crack captcha. Do you provide a captcha decoders?

> DeCaptcher CAPTCHA solving is processed by humans. So the accuracy is much better than an automated captcha solver ones