**Motivation**

Arrgh! There is so much bad traffic on the internet!
- DoS attacks
- Worms
- Port scans
- Spambots
- Spoofer attacks
- Game cheaters

**Question:** What can be done?

Client puzzles offer an ideal punishment mechanism:
- Easy to assign punishment
- Can make punishment arbitrarily difficult
- False positives degrade but do not deny service

Other work secures individual protocol vulnerabilities, however the most effective solution should protect all network traffic; thus it must be placed in the IP layer.

**Our approach:**
- **IP layer client puzzles**

---

**Challenges**

**Flexible Deployment**
- Puzzle issuers at arbitrary network locations

**Minimal Overhead**
- Puzzles can be generated at line speed
- Constant state at the puzzle issuer
- Minimal packet expansion

**Tamper Resistance**
- Replay attacks
- Spoofer attacks
- Work ahead attacks

**Support for Real Time Apps**
- Online games
- Streaming media

---

**Puzzle Protocol**

<table>
<thead>
<tr>
<th>Client</th>
<th>Issuer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Cookie</td>
<td>Server Cookie, F, Puzzle Parameters</td>
</tr>
<tr>
<td>Client Cookie, Server Cookie, Answer</td>
<td></td>
</tr>
</tbody>
</table>

**Puzzle Algorithm**

**Hint-Based Hash-Reversal**

**Requirements:**
- Keyed HMAC (h)
- High entropy random number generator (rand)

**Creating the Puzzle:**
1. Answer = rand()
2. Hint = Answer - (rand() mod Difficulty)
3. Puzzle Hash = (Answer)
4. discard the Answer

**Solving the Puzzle:**
1. Search Value = Hint
2. if (Search Value = Puzzle Hash) then: Answer = Search Value
3. Search Value = Search Value + 1
4. go to step 2

---

**Protocol Extensions**

**IP Options Cookie:**
- Issue Timestamp
- Client Nonce
- F
- Server Cookie
- Server Port
- Protocol
- Maturity Time
- Puzzle Difficulty
- Puzzle Parameters (variable length)

**ICMP Puzzle:**
- Type = 26
- Code = 0
- Server IP
- Client Port
- Server Port
- Maturity Time
- Puzzle Difficulty
- Puzzle Parameters (variable length)

---

**iptables Implementation**

**Puzzle Proxy**
- Cache Packet
- Add Cookie to IP Header
- Retransmit Packet
- Solve Puzzle
- Add Answer to IP Header

**Puzzle Firewall**
- Need Puzzle?
- No
- Issue ICMP Puzzle
- Drop Packet
- Valid Answer?
- No
- Next Packet
- Yes

---

**Performance**

**Constant State at Issuer**
- Fast to issue
  - Requires only one hash and two random numbers

**Fine Grain Difficulty Control**
- Can linearly increment puzzle difficulty

**Throughput**
- Tests use:
  - Dual 1.8GHz Intel Xeon machines
  - Cisco Catalyst 4000 Gigabit switch

- Firewall:
  - Validate and issue puzzles at 182,000 packets/s
  - Proxy:
    - Solve min-difficulty puzzles at 130,000 packets/s
    - Solve max-difficulty puzzles at << 1 packets/s

---

**Slowing Port Scans**

**Time to Scan (s)**

- As difficulty increases, time to scan increases

---

**Future Work**

**Reputation-Based Networking**
- Keep interaction history about clients
- Determine their reputability
- Use IP Puzzles to punish clients who are bad
- Share knowledge with other IP Puzzle firewalls

**Publicly Auditable Puzzles**
- Puzzle answers can be independently verified by intermediate IP Puzzle routers
- Answers can indicate amount of work done

**Puzzles With Useful Answers**
- Puzzle algorithms where the answers provide useful computation for the puzzle issuer
- Puzzle answers must be easily verifiable

---

**OGI SCHOOL OF SCIENCE & ENGINEERING**

**OREGON HEALTH & SCIENCE UNIVERSITY**

http://www.cse.ogi.edu/sysl/

**Funded by:**

intel