Not-a-Bot (NAB): Improving Service Availability in the Face of Botnet Attacks

Ramakrishna (Ramki) Gummadi
MIT

Hari Balakrishnan (MIT), Petros Maniatis and Sylvia Ratnasamy (Intel Research)
The problem: Service unavailability

Subject: Mail delivery failed: returning message to sender
From: Mail Delivery System <Mailer-Daemon@csail.mit.edu>
Date: 4/21/2009 7:56 AM
To: ramki@csail.mit.edu

This message was created automatically by mail delivery software.

A message that you sent could not be delivered to one or more of its recipients. This is a permanent error. The following address(es) failed:

voelker@cs.ucsd.edu
SMTP error from remote mail server after end of data:
host cse-inbound.ucsd.edu [132.239.51.75]: 554 Service unavailable; Client host [outgoing.csail.mit.edu] blocked using Barracuda Reputation; http://bbl.barracudacentral.com/q.cgi?ip=194.209.131.192

Schneier's Crypto-Gram is getting flagged as spam by Razor. The reason is that some spam-detecting software will try to automatically detect spam and then automatically report it. So somebody's SpamAssassin mistakenly concludes that a copy of Crypto-Gram is spam and reports it to Razor, and this happens a few times over; now everyone who uses Razor will automatically be advised that Razor considers Crypto-Gram to be spam!
Botnets: Reduce service availability

Email: 85% of spam from top six botnets
- Over 95% of all inboxes affected
- 120 billion messages/day: Overloaded mail servers

DDoS

Question: General way to distinguish bots from humans?

Click-fraud: ad fraud, search engine fraud
- ~15% of all ad clicks
- Compromise search results
Existing solutions

**CAPTCHAs**

Drawback: Intrusive

**User Account Control**

Drawback: Default “yes” [Whitten, Tygar ’99]

How to distinguish humans from bots automatically?
Strawman: Attesting human activity with Trusted Platform Modules

<table>
<thead>
<tr>
<th>Keystrokes</th>
<th>Attested Keystrokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Images</td>
</tr>
<tr>
<td>tom cruise</td>
<td>6,670,000 results</td>
</tr>
<tr>
<td>tom cruise</td>
<td>1,990,000 results</td>
</tr>
<tr>
<td>tom clancy</td>
<td>2,670,000 results</td>
</tr>
<tr>
<td>tom cruise movies</td>
<td>215,000 results</td>
</tr>
<tr>
<td>tom cruise height</td>
<td>5,780,000 results</td>
</tr>
<tr>
<td>tom collins</td>
<td>145,000 results</td>
</tr>
<tr>
<td>tom cruse</td>
<td>730,000 results</td>
</tr>
<tr>
<td>tom clancy books</td>
<td>481,000 results</td>
</tr>
<tr>
<td>tom cruise filmography</td>
<td>552,000 results</td>
</tr>
<tr>
<td>tom chaplin</td>
<td>347,000 results</td>
</tr>
<tr>
<td>tom cochrane</td>
<td></td>
</tr>
</tbody>
</table>
Problems with the strawman

- Google Suggest

<table>
<thead>
<tr>
<th>Web</th>
<th>Images</th>
<th>Groups</th>
<th>News</th>
<th>Froogle</th>
<th>Local</th>
<th>more</th>
</tr>
</thead>
<tbody>
<tr>
<td>tom cruise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom cruise</td>
<td>6,670,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom clancy</td>
<td>1,990,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom cruise movies</td>
<td>2,670,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom cruise height</td>
<td>215,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom collins</td>
<td>5,780,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom cruse</td>
<td>145,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom clancy books</td>
<td>730,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom cruise filmography</td>
<td>481,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom chaplin</td>
<td>552,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tom cochrane</td>
<td>347,000 results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Attested Keystrokes
- Browser
- OS
- High-rate clicks
- Slow

NSDI 2009
Assumptions and Requirements

Assumptions

- Untrusted OS
- Verifiable TPM bootup
- Correct implementation of cryptographic primitives

Requirements

- Automatic
- Fast (handle interactive traffic)
- Small TCB (Trusted Computing Base)
- Preserve privacy and anonymity
TPM Background

- Small, physically sealed chip
- Internal private key for measuring and reporting system integrity
- Two relevant protocols
  - Direct anonymous attestation
    - Group signatures using a key $K_{priv}$
  - Sealed storage
    - Secure location to store $K_{priv}$ until system integrity verified
NAB (Not-A-Bot) Architecture

! Goal: Attest all human requests, reduce attested bot requests

•! No blacklisting: human requests from compromised hosts still receive service
Attestation security properties

- Non-transferable
  - Cannot generate at one host, use at another
- Bound to request content
- No way to send spam from bots using one gmail account
- Single-use (verifier detects dupes)
- Limited valid time-window
When to attest?

![Simple, timing-based attestation](Image)
- Requires human activity
- Allow attestation when request received within $\delta_{\{k,m\}}$ of last keyboard, mouse click
- Attester provides attestation only if $\delta_{\{k,m\}} < \Delta_{\{k,m\}}$ (= 1s for email)
  - Verifier checks $\delta_{\{k,m\}}$ in attestation for validity
- Reduces click harvesting
What to attest?

Challenger-specific
- Cannot be retargeted

Responder-specific
- Cannot exploit manually configured whitelisting

Content-specific
- Cannot modify or piggyback on valid messages
What is in an attestation?

- Signed SHA-1 hash of message
- 160-bit signed nonce
  - Verifier stores nonces for application-defined period, checks duplicates
- Optional $\delta_{\{k,m\}}$ values (omitted for privacy)
- Certificate to verify $K_{priv}$

| Attestation | $K_{priv}\{H(msg)\}$ | Siged Nonce | $K_{priv}\{\delta_m, \delta_k\}$ | certified $K_{pub}$ |
Attester Interfaces

req(h(msg), type, PID)

User → kbd, mouse clicks

Measure integrity, release certified {K_{pub}, K_{priv}} at boot

TPM

Attestation

App

OS

Type: Anonymous or non-anonymous

PID: Delayed attestation release for a process

NSDI 2009
Attester Operation

Installation: Set to use TPM register # 18:
\[ PCRE\text{extend}(18, \text{Hash(Attester Code)}) \]

Sealing private key \( K_{\text{priv}} \) to host:
\[ S = \text{Seal}(18, K_{\text{priv}}) \]

Booting: Release \( K_{\text{priv}} \) to attester:
\[ K_{\text{priv}} = \text{Unseal}(S, (18, \text{PCR}_{18})) \]

Recomputed attester’s hash
Verifier Operation

- Checks validity of $K_{priv}$, attestation, nonce
- Uses application-specific policies

Email:

```
mail

Below spam assassin threshold?
  yes
  no

  Forward

Attested?
  yes
  no

  Nonce valid?
    yes
    no

    Forward

  Discard

  Discard
```
Email: Usage scenarios and incentives

Mailing lists
- Verifier checks subscription to mailing list name in “To:” field

Offline mode
- Attestation requested when user hits “send”

Sender incentive
- Better email reliability

Recipient incentive
- Reduced mail server load, better reliability
Request processing at verifier

Prioritize attested requests

Attested

High priority

Requests

Unattested

Low priority

Overloaded email, web server

NSDI 2009
DDoS, Click-fraud: Usage and incentives

! Browser gets attestation when requesting document root (“http://foo.com/”)
•! Verifier stores attestation, accepts same attestation in future for all embedded links
•! 10 minutes expiry

!Browser forced to use new attestation for next fetch

!Incentive: Attester distributed in search engine toolbars
Evaluation

! Implemented attester with Xen VMM
  • Uses domain disaggregation [Murray et al.,’08]
  • Attester within a paravirtualized Xen domain built with miniOS, isolated from untrusted OS

! Trace-driven verifier evaluation
  • Click traces of 328 users in one month [Giroire et al.,’08]
  • Publicly available spam, DDoS and click-fraud traces
  • Worst-case scenario with adaptive bots
Attester evaluation

CPU cost: At most 10 ms on 2 GHz CPU
- RSA signatures, 1024-bit modulus

Complexity metric: lines of code
- Attester kernel module: 500 lines
- miniOS: 30,000 lines

Applications: NET::SMTP (Email), cURL (Web)
- 250 lines of code modified
- Attestations as extended protocol objects
Verifier evaluation

! Methodology: 328 click traces at 1s intervals
  • ! Adaptive bot: steals as many clicks as possible
  • ! Generates traffic using all stolen clicks
  • ! Compare against status quo (normal bot without NAB) within the same time
  • ! 328 data points, one for each user’s trace

! Other metrics
  • ! Nonce storage cost (< 600 GB for one-month nonces with million clients)
  • ! Throughput: 10,000 attestations/s
Spam mitigation

Default: 1.5% missed spam, 0.08% misclassified as spam

NAB: 0.15% missed spam, 0% misclassified as spam

NAB reduces inbox spam by 90%
Email server overload mitigation

No trace sees more than 8% prioritized spam

NAB reduces email server overload by at least 92%
DDoS mitigation

No trace sees more than 11% prioritized DDoS

NAB mitigates 89% of DDoS requests
Click-fraud mitigation

No trace sees more than 13% click-fraud traffic

NAB reduces click-fraud by 87%
Related work

- Human activity detection
  - CAPTCHAs [Ahn et al.,’03]
    - Susceptible to man-in-the-middle attack
  - Nexus [Williams et al.,’08]
    - Not for commodity OSes

- Mitigating spam, DDoS, click-fraud
  - Spam: Occam [Fleizach et al.,’07], SPF, DKIM
  - DDoS: Path validation, bandwidth-as-payment
  - Click-fraud: Syndicators, clickable CAPTCHAs
  - Mostly specialized, share little commonality
Conclusions

NAB: Improves service availability in the presence of botnets
- Even on botted hosts, users get ~ 100% service
  - No blacklisting
- De-prioritize or drop up to 90% bot traffic

Automatic content- and machine-specific attestations

Single abstraction, support for application-specific verifier policies

Future work: Attestation without virtualization