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

Bounced email

Misclassified email

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: 4000 attacks/week [Moore et al., '06]

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

Question: General way to distinguish bots from humans?
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
Problems with the strawman

Attested Keystrokes

Browser
OS

Slow

High-rate clicks

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
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
  - 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

To: bob@b.org
From: alice@a.org
Hi Bob,…
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 $K_{\text{priv}}$

| Attestation | $K_{\text{priv}}\{H(\text{msg})\}$ | Signed Nonce | $K_{\text{priv}}\{\delta_m, \delta_k\}$ | certified $K_{\text{pub}}$ |
Attestor Interfaces

User

kbd, mouse clicks

Attestor

req(h(msg), type, \(\phi_m, \phi_k, \text{PID}\))

Attestation

TPM

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

Type: Anonymous or non-anonymous

PID: Delayed attestation release for a process
Attester Operation

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

- 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 $K_{\text{priv}}$, attestation, nonce
- Uses application-specific policies
- Email:

```
mail

Below spam assassin threshold?

- yes: Forward
- no: Attested?

Attested?

- yes: Nonce valid?
- no: Discard

Nonce valid?

- yes: Forward
- no: 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

Requests

Attested

High priority

Unattested

Low priority

Overloaded email, web server

Prioritize attested requests
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
  - Atester 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