Not-A-Bot: Improving Service Availability in the Face of Botnet Attacks

R. Gummadi, H. Balakrishnan, P. Maniatis, S. Ratnasamy

Presented by: Ashish Vulimiri

Images lifted from paper/authors' NSDI09 slides. All hail the fair use exception.

Motivation

Motivation

- Botnets: bad

Motivation

- Botnets: bad
  - Spam
  - DDoS
  - Click-fraud

Motivation

- Botnets: bad
  - Spam
  - DDoS
  - Click-fraud
- Problem: cannot distinguish bot/human requests

Motivation

- Botnets: bad
  - Spam
  - DDoS
  - Click-fraud
- Problem: cannot distinguish bot/human requests
- Will solving this issue always help?

Related Work

- Application-specific schemes
  - Bandwidth/computation based payment schemes for DoS
  - Sender authentication schemes like SPF, DomainKeys for spam control
- Human-activity detection
  - CAPTCHAs

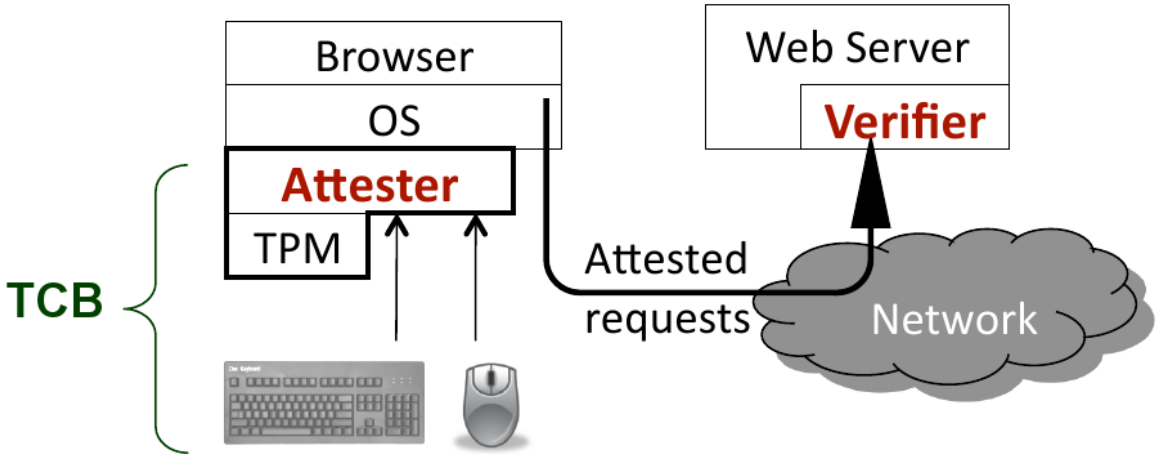
Related Work

- Application-specific schemes
  - Bandwidth/computation based payment schemes for DoS
  - Sender authentication schemes like SPF, DomainKeys for spam control
- Human-activity detection
  - CAPTCHAs
- Secure execution environments
  - Pioneer
  - XOM
  - TPM, vTPM

TPM

- Trusted base
- Cryptographic coprocessor
- Not-A-Bot uses:
  - Platform configuration registers
  - Sealed storage
    - Can seal values, signed by TPM's internal key, along with guard conditions on the value of PCRs
  - Direct anonymous attestation

Not-A-Bot



- Chain of trust from attester to verifier
- When requested, attester checks and signs off on human originated actions
- Guaranteed human requests can be given higher priority at server
- Granularity is request level, not host level – human requests from compromised hosts might benefit

Chain of Trust

Chain of Trust

- PCRs are used to provide verifiable bootup

Chain of Trust

- PCRs are used to provide verifiable bootup
- When attester is installed, private information sealed using TPM, with BIOS and attester code hashes as guards. Private info includes:
  - Private key κ_{priv}
  - Information needed to create a signed certificate for DAA. This is NOT a shared secret

Chain of Trust

- PCRs are used to provide verifiable bootup
- When attester is installed, private information sealed using TPM, with BIOS and attester code hashes as guards. Private info includes:
  - Private key κ_{priv}
  - Information needed to create a signed certificate for DAA. This is NOT a shared secret
- TPM allows unsealing only if BIOS and attester hashes match – so if attester code is changed, key can't be accessed

Chain of Trust

Chain of Trust

- Application (at client) must request attestation locally from the attester and send to verifier to authenticate that a request is human-generated

Chain of Trust

- Application (at client) must request attestation locally from the attester and send to verifier to authenticate that a request is human-generated
- An attestatation is of the form $\langle a, sign(\kappa_{priv}, a), C \rangle$, where a is the attestation information and C is a certificate that attester uses with the DAA protocol to prove integrity to the verifier

Chain of Trust

- Application (at client) must request attestation locally from the attester and send to verifier to authenticate that a request is human-generated
- An attestatation is of the form $\langle a, sign(\kappa_{priv}, a), C \rangle$, where a is the attestation information and C is a certificate that attester uses with the DAA protocol to prove integrity to the verifier
- Necessary component of a: nonce n, which the verifier stores to ensure client is not replaying authentications

Attester Operation

- Request is considered human-generated if it occurs within $\langle \Delta_m, \Delta_k \rangle$ distance of a mouse/keyboard click, where the Δ parameters are application specific
- Attestation may either include time since last mouse click/keypress directly, or merely state an upper-bound on them (the first leaks some timing information which may be significant)
- Choice left to application
- Attestation information a is $\langle d, n, \delta_m, \delta_k \rangle$, where d is a digest of the message (e.g. e-mail, HTTP GET/POST etc), n is the nonce used to ensure client cannot replay attestations, δ is timing information

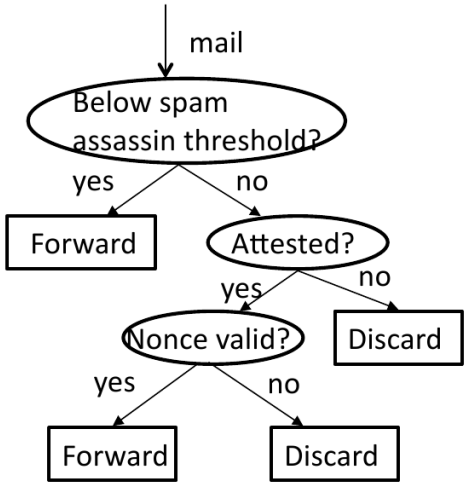
Verifier Operation

Spam

- In attestation, entire message is hashed: including sender, recipient, timestamp and content
- Server stores nonces for a month
- Together, these two factors severely restrict replayability: spammer can reuse authentication only after a month (only one replay per authenticated email)
- But because timestamp is also hashed, it can't be changed. Server will reject even this lone replayed email as too old.

Verifier Operation

Spam



Additional notes:

- For mailing lists, auth sent to each email address in the “To:” field
- Offline mode: store an auth when user clicks “Send”, hold it until connected to the network
- Script mode: similar to offline mode. User manually authorizes a certain number of human-authentications when writing a script

Verifier Operation

DDoS/Click Fraud

- Browser sends authentication for document root (e.g. “http://www.example.com/”)
- Server stores auth for 10 minutes
- In this time, the authentication also grants access to any embedded links/documents
- Note: unlike with e-mail, incentive structure is asymmetric. Much more useful to website owners/content providers than to users

Verifier Operation

DDoS/Click Fraud

- Browser sends authentication for document root (e.g. “http://www.example.com/”)
- Server stores auth for 10 minutes
- In this time, the authentication also grants access to any embedded links/documents
- Note: unlike with e-mail, incentive structure is asymmetric. Much more useful to website owners/content providers than to users
  - Authors suggest that verifiers push attesters onto users through other means, for example browser toolbars

Experimental Evaluation

- Spam
  - Client: reduced false negatives in inbox from 1.5% to 0.15%, false positives from 0.08% to 0%
  - Server: of all spam traffic, 8% was attested as human-originated
- DDoS
  - 11% of all DDoS requests attested as human-originated
- Click-fraud
  - 13% of all click-fraud traffic attested as human-originated

Discussion

Discussion

- What else (apart from non-human origin) characterizes botnet requests?

Discussion

- What else (apart from non-human origin) characterizes botnet requests?
- Better human-identification algorithm?

Discussion

- What else (apart from non-human origin) characterizes botnet requests?
- Better human-identification algorithm?
- How reasonable is it to assume hardware safety?

Discussion

- What else (apart from non-human origin) characterizes botnet requests?
- Better human-identification algorithm?
- How reasonable is it to assume hardware safety?
- Trusted computing issues

Questions?