ECS 235B Module 24
Trust Models
Trust Models

• Integrity models state conditions under which changes preserve a set of properties
  • So deal with the *preservation* of trustworthiness

• Trust models deal with confidence one can have in the initial values or settings
  • So deal with the *initial* evaluation of whether data can be trusted
Definition of Trust

Anna *trusts* Bill if Anna believes, with a level of subjective probability, that Bill will perform a particular action, both before the action can be monitored (or independently of the capacity of being able to monitor it) and in a context in which it affects Anna’s own action.

- Includes subjective nature of trust
- Captures idea that trust comes from a belief in what we do not monitor
- Leads to transitivity of trust
Transitivity of Trust

*Transitivity of trust*: if Anna trusts Bill and Bill trusts Carole, then Anna trusts Carole

- Not always; depends on Anna’s assessment of B’s judgment

- *Conditional transitivity of trust*: A trusts C when
  - B recommends C to A;
  - A trusts B’s recommendations;
  - A can make judgments about B’s recommendations; and
  - Based on B’s recommendation, A may trust C less than B does

- *Direct trust*: A trusts C because of A’s observations and interactions

- *Indirect trust*: A trusts C because A accepts B’s recommendation
Types of Beliefs Underlying Trust

• Competence: A believes B competent to aid A in reaching goal
• Disposition: A believes B will actually do what A needs to reach goal
• Dependence: A believes she needs what B will do, depends on what B will do, or it’s better to rely on B than not
• Fulfillment: A believes goal will be reached
• Willingness: A believes B has decided to do what A wants
Evaluating Arguments about Trust (con’t)

- **Persistence**: A believes B will not change B’s mind before doing what A wants
- **Self-confidence**: A believes that B knows B can take the action A wants
- **Majority behavior**: A’s belief that most people from B’s community are trustworthy
- **Prudence**: Not trusting B poses unacceptable risk to A
- **Pragmatism**: A’s current interests best served by trusting B
Trust Management

• Use a language to express relationships about trust, allowing us to reason about trust
  • Evaluation mechanisms take data, trust relationships and provide a measure of trust about the entity or whether an action should or should not be taken

• Two basic forms
  • Policy-based trust management
  • Reputation-based trust management
Policy-Based Trust Management

• Credentials instantiate policy rules
  • Credentials are data, so they too may be input to the rules
  • Trusted third parties often vouch for credentials

• Policy rules expressed in a policy language
  • Different languages for different goals
  • Expressiveness of language determines the policies it can express
Example: Keynote

- **Basic units**
  - **Assertions**: describe actions allowed to possessors of credentials
    - **Policy**: statements about policy
    - **Credential**: statements about credentials
  - **Action environment**: attributes describing action associated with credentials
- **Evaluator**: takes set of policy assertions, set of credentials, action environment and determines if proposed action is consistent with policy
Example

- Consider email domain: policy assertion authorizes holder of mastercred for all actions:
  
  Authorizer: "POLICY"
  Licensees: "mastercred"

- Credential assertion:
  
  KeyNote-Version: 2
  Local-Constants: Alice="cred1234", Bob="credABCD"
  Authorizer: "authcred"
  Licensees: Alice || Bob
  Conditions: (app_domain == "RFC822-EMAIL") &&
  
  (address =~ "^.*@keynote\.ucdavis\.edu$")
  Signature: "signed"

- Compliance Value Set: { "_MIN_TRUST", "_MAX_TRUST" }
Example: Results

• Evaluator given action environment:
  
  _ACTION_AUTHORIZERS = "Alice"
  app_domain = "RFC822-EMAIL"
  address = "snoopy@keynote.ucdavis.edu"

  it satisfies policy, so returns _MAX_TRUST

• Evaluator given action environment:
  
  _ACTION_AUTHORIZERS = "Bob"
  app_domain = "RFC822-EMAIL"
  address = "opus@admin.ucdavis.edu"

  it does not satisfy policy, so returns _MIN_TRUST
Example 2

• Consider separation of duty: policy assertion delegates authority to pay invoices to entity with credential “fundmgrcred”:
  
  Authorizer: "POLICY"
  Licensee: "fundmgcred"
  Conditions: (app_domain == "INVOICE" && @dollars < 10000)

• Credential assertion (requires 2 signatures on any expenditure):
  
  KeyNote-Version: 2
  Comment: This credential specifies a spending policy
  Authorizer: "authcred"
  Licensees: 2-of("cred1", "cred2", "cred3", "cred4", "cred5")
  Conditions: (app_domain=="INVOICE") # note nested clauses
  -> { (@dollars) < 2500) -> "Approve";
      ( @dollars < 7500) -> "ApproveAndLog"; }

  Signature: "signed"

• Compliance Value Set: {"Reject", "ApproveAndLog", "Approve" }
Example 2: Results

• Evaluator given action environment:
  
  _ACTION_AUTHORIZERS = "cred1,cred4"
  app_domain = "INVOICE"
  dollars = "1000"

  it satisfies first clause of condition, and so policy, so returns Approve

• Evaluator given action environment:
  
  _ACTION_AUTHORIZERS = "cred1"
  app_domain = "INVOICE"
  dollars = "1500"

  it does not satisfy policy as too few Licensees, so returns Reject
Example 2: Results

• Evaluator given action environment:
  
  \[
  \begin{align*}
  \_\text{ACTION\_AUTHORIZERS} & = "\text{cred1,cred2}" \\
  \text{app\_domain} & = "\text{INVOICE}" \\
  \text{dollars} & = "3541"
  \end{align*}
  \]

  it satisfies second clause of condition, and so policy, so returns \text{ApproveAndLog}

• Evaluator given action environment:

  \[
  \begin{align*}
  \_\text{ACTION\_AUTHORIZERS} & = "\text{cred1,cred5}" \\
  \text{app\_domain} & = "\text{INVOICE}" \\
  \text{dollars} & = "8000"
  \end{align*}
  \]

  it does not satisfy policy as amount too large, so returns \text{Reject}
Reputation-Based Trust Management

• Use past behavior, information from other sources, to determine whether to trust an entity

• Some models distinguish between direct, indirect trust

• Trust category, trust values, agent’s identification form reputation

• Recommendation is trust information containing at least 1 reputation

• Systems use many different types of metrics
  • Statistical models
  • Belief models (probabilities may not sum to 1, due to uncertainty in belief)
  • Fuzzy models (reasoning involves degrees of trustworthiness)
Example 1

- Direct trust: $-1$ (untrustworthy), 1 to 4 (degrees of trust, increasing), 0 (cannot make trust judgment)
- Indirect trust: $-1, 0$ (same as for direct trust), 1 to 4 (how close the judgment of recommender is to the entity being recommended to)
- Formula:

$$ t(T, P) = tv(T) \prod_{i=1}^{n} \frac{tv(R_i)}{4} $$

where $T$ is entity of concern, $P$ trust path, $tv(x)$ trust value of $x$, $t(T,P)$ overall trust in $T$ based on trust path $P$
Example 1

• Amy wants Boris’ recommendation about Danny so she asks him
  • Amy trusts Boris’ recommendations with trust value 2 as his judgment is somewhat close to hers
• Boris doesn’t know Danny, so he asks Carole
  • He trusts her recommendations with trust value 3
• Carole believes Danny is above average programmer, so she replies with a recommendation of 3
• Boris adds this to the end of the recommendation
• \( P \) is Amy—Boris—Carole—Danny, so \( R_1 = \text{Boris}, R_2 = \text{Carole}, T = \text{Danny}, \) so

\[
T("\text{Danny}, P") = 3 \times \frac{2}{4} \times \frac{3}{4} = 1.125
\]
Example 2

- PeerTrust uses metric based on complaints
- $u \in P$ is a node in a peer-to-peer network
- $p(u, t) \in P$ is node that $u$ interacts with in transaction $t$
- $S(u, t)$ is amount of satisfaction $u$ gets from $p(u, t)$
- $I(u)$ is total number of transactions
- Trust value of $u$: $T(u) = \sum_{t=1}^{I(u)} S(u, t) Cr(p(u, t))$
- Credibility of node $x$’s feedback: $Cr(x) = \sum_{t=1}^{I(x)} S(x, t) \frac{T(p(x, t))}{\sum_{y=1}^{I(x)} I(x) T(p(x, y))}$
- So credibility of $x$ depends on prior trust values
Key Points

• Integrity policies deal with trust
  • As trust is hard to quantify, these policies are hard to evaluate completely
  • Look for assumptions and trusted users to find possible weak points in their implementation

• Biba based on multilevel integrity

• Clark-Wilson focuses on separation of duty and transactions