NTIA Meeting Software Transparency

- The purpose of this presentation is to help ensure transparency proposals are evaluated on both positive and negative outcomes
- Agenda
  - Defining goals for different constituencies interested in software transparency
  - Consider possible negative effects of transparency

Define Goals for Different Constituencies

- Goals need to be defined first, in order to evaluate proposals, pro and con
  - Goals should be defined for different constituencies (Vendors, Customers, 3rd party component developers)
- Proposed goals (and non goals) can be high level and/or low level
  - High: Improve production product security while maintaining or reducing disruptions
  - Low: Produce an inventory of fixes for 3rd party components that are only exploitable within the context of the Vendor product and customer production deployment
  - High: Don't provide gratuitous information to avoid unintended, bad consequences
    - E.g. Sales calls from other 3rd parties, boycotts for 3rd party "bad" vendors, SAML sales gambits, …
- Consider a large but not among the largest of Oracle products
  - 300+ 3rd party components
  - 150+ "Vendors" of 3rd party components
  - 25% of "Vendors" are named individuals

Considering Possible Goals **and** Outcomes

- Vendor goals might include software re-creation plus customer goals
- Customer goals might include faster production inclusion of exploitable vulnerability fixes either for security **or** compliance reasons or both
- 3rd party dev goals might be to prevent onerous regulations that inhibit 3rd party component use (i.e. effect may be to inhibit use of 3rd party code)
- Consider auxiliary requirements
  - Need 3rd party component vendor, product and patch unique IDs with aliases
  - Need structured formats suitable for efficient automation

Consider Possible Negative Effects of Transparency

- Negatives: Actually reducing security or increasing production disruption
- Current there are negatives caused by increased transparency
  - Customers using tools to construct 3rd party inventories & patches needed per NVD
  - **Question**: Are these tools good enough now (or is this expected soon?)
  - Considerable customer "mandated" patching without benefit is occurring now
    - Log4j recent fix: Hundreds of products updated, less than five actually exploitable
      Customers: Demand patches even with not-exploitable vendor claims (e.g. because of compliance)
      Result: 1000's of customers disrupted with no security benefit, and possible security degradation
    - Heartbleed: Hundreds of products fixed but only 20 were exploitable (most used crypto only)
- For many products, patching within one week requires multiple fixes/week
  - Incompatible with production for large products, customers will pick and choose
    Customers can't pick and choose effectively because they lack information
    Result: Customer security degraded versus fixed, scheduled, patch sets
    - Make sure the side effect of the "fix" isn't worse than the problem