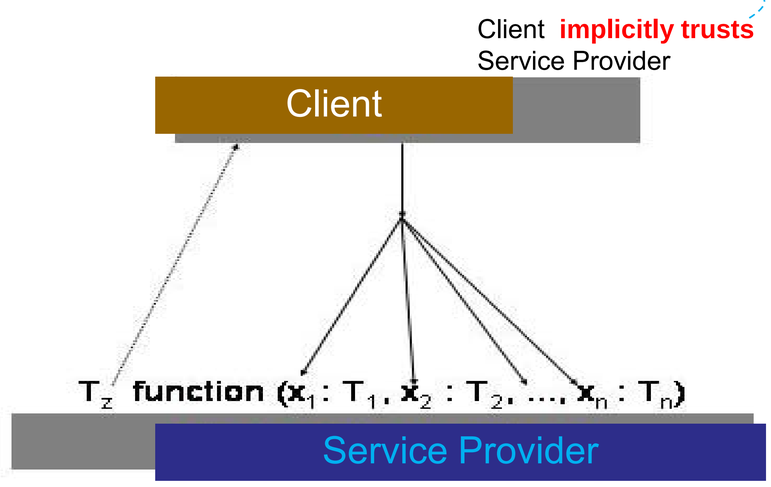
CERIAS

the center for education and research in information assurance and security

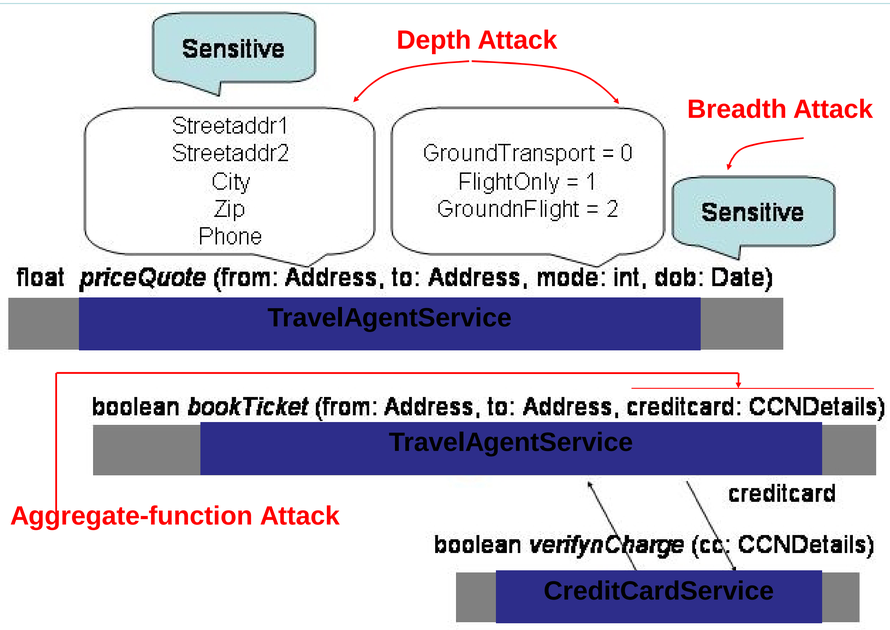
A Serious Form of Attack: SN2K - Software-based Need-to-Know Attacks

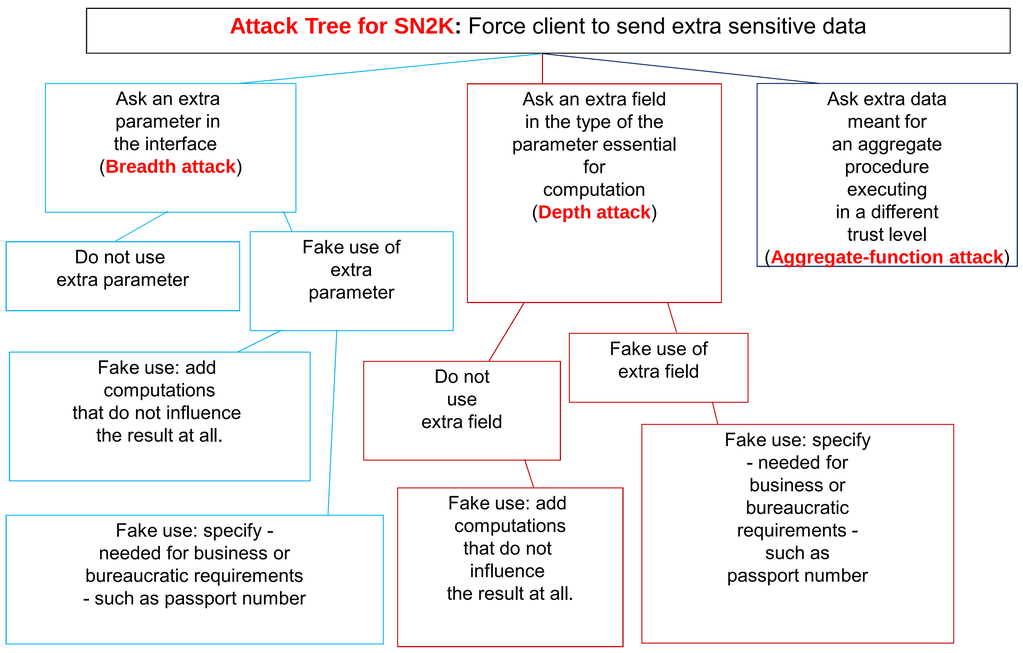
Ashish Kundu, CS & CERIAS, Purdue University (ashishk@cs.purdue.edu)

Existing Programming Models: Insecure



Example





Feasibility of Attack
Simple
Efficient
Asymmetric cost to client (more) and to attacker (very less)

Language Mechanisms for Attack
Polymorphism: depth attack
Aliasing: fake use
Undecidability: fake use

Sources of Attack
Malicious service providers
Software evolution
Automatic/Dynamic service composition
Malicious insider

Dimensions of a Solution
Type-safety
Static or dynamic typing
Alias analysis
Static program analysis
Dynamic program analysis
Semantic analysis

SN2K Attacks: Types

Depth Attack (1)	at least one field in a composite parameter
Breadth Attack (2)	at least one parameter
Aggregate-function Attack (3)	at least one parameter/field declared by an aggregate, but used only by a component function

not necessary for the service offered

Formal Semantics:

Function f: R = {p | p = (T, x)}, field f Є T, T_1 = T – {f}.

(1) R' = R U (T_1, x) – {p} [f]R' = [f]R

(2) R'' = R – {p}, [f]R'' = [f]R

(3) Combination of (1) and (2) in aggregate functions/services.

Detecting an Attack and Certification

Input: Function f: *signature (S)*, *implementation*(I)

Signature: *RetType f (T_1: x_1, T_2: x_2, … T_n: x_n)*

1. Alias analysis: determine aliases in I
2. Dead code elimination on I: generate I' with no dead code
3. Type-inferencing on I'
   1. detect parameters used in *depth attack* and *breadth attack*
   2. **prune** these parameters to generate S' from S
4. Forward slicing from S' over I': params affecting result
5. Interprocedural semantic analysis on I': program properties, invariants, aggregation-function attack
6. **Prune** params/computations that do not affect result (4, 5): I''
7. Carry out dynamic analysis, to remove fake use, if any detected: S''', I'''
8. **Certify S''' and I''': *signed-hash(S''', binary of I''').***

Fake use

Never used

Fake use

Future directions: non-monopolistic programming model.

Reference: Software-based Need-to-know attacks (SN2K attacks), Ashish Kundu, to be submitted.

PURDUE UNIVERSITY

CERIAS

Discovery Park e-Enterprise Center