Operating System Discovery Using Answer Set Programming

François Gagnon
Ph.D Student
Carleton University
fgagnon@sce.carleton.ca
www.sce.carleton.ca/~fgagnon

What is Operating System Discovery?

- Remotely identifying which operating systems are running on distant computers
- Ex.: Windows 200 server SP2, Linux 2.2.7, FreeBSD 5.0, etc
- Using peculiarities in TCP/IP stack implementation caused by ambiguities in the protocol specifications
- Ex: How to fill the destination MAC field of an ARP request
  - FF:FF:FF:FF:FF:FF — sun and mac prior to version 10
  - Random data — FreeBSD 4.6, 4.7, 4.8, and 5.0
  - 00:00:00:00:00:00 — every other
- Knowing the operating system is useful to determine if a machine is vulnerable to a given attack.

Current OSD Approaches

Passive:

- Lack knowledge representation:
  - One guess per packet
  - No memory of previous guess
  - No stimulus-response correlation
- Limited to the information they receive
- Limited accuracy

Active:

- Lack knowledge representation:
  - Redo the work for each query
  - No memory of previous test results
- Lack planning ability
  - Always run all tests
  - Very noisy
- Don't use the information freely available

Answer Set Programming

- Extended disjunctive logic programs with answer set semantics

$$L_1 \vee \ldots \vee L_k \leftarrow L_{k+1}, \ldots, L_m, \mathit{not}\ L_{m+1}, \ldots, \mathit{not}\ L_n$$

  - Where each L_i is a literal (A) or its strong negation (¬A)
  - *not* denotes weak negation
- A set of ground literal S is an answer set of program ∏ if:
  - the literals of S are those made true by ∏
  - the literals of S are sufficient to respect the rules of ∏
  - no proper subset of S is also an answer set
- A program may have multiple answer sets
- The language is fully declarative (can be generated automatically)

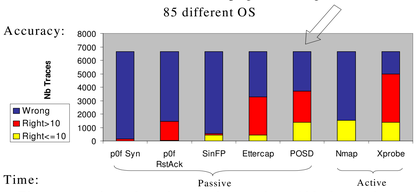
Passive Module (POSD)

- Represented as an explanatory diagnosis problem <Hyp,Th,Obs>:
  - Hyp: the set of currently possible explanations (OS)
  - Th: a set of rules describing the behavior of each OS
  - Obs: the packets seen so far
- Example of a behavior rule:
  os(win2K) v os(winXP) ← tcp(IpS,IpD,PS,PD,yes, syn, 128).
- Each answer set provides a possible explanation (OS)
  - We prioritize smaller answers since they are more general
- If no answer set contains only 1 OS, it could mean:
  - The target changed its OS
  - The target is actually multiple computers behind a NAT
- The observations non-monotonically confirm some hypotheses
- The logic program is generated automatically

Benchmarks (POSD)

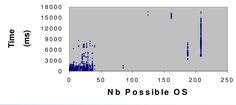
6671 attack traces containing up to 63000 packets
85 different OS

- Accuracy:



- Time:
  - The program is evaluated every time we have 100 observations
  - Observations are discarded (memorized through the explanations)

avgTime: 386 ms
minTime: 102 ms
maxTime: 16123 ms



Active Module (Future Work)

- Given a set of hypotheses H, generated by the passive module, we can ask several queries:
- Is O the actual operating system?
  - *yes* if H = {O}
  - *no* if O ∉ H
  - *unknown* otherwise
- Does the actual operating system belong to θ?
  - *yes* if θ ⊇ H
  - *no* if θ ∩ H = ∅
  - *unknown* otherwise
- What is the actual operating system?
  - h if H is a singleton H = {h}
  - *unknown* otherwise
- Generate a (conditional) plan to gather the missing observations

CARLETON university