Attack Planning in the Real World

Jorge Lucángeli Obes[1] Carlos Sarraute [1,2]

[1] CoreLabs - Core Security Technologies

[2] ITBA (Instituto Tecnológico de Buenos Aires)

SecArt'10 - July 12, 2010

Introduction

Our company: Core Security Technologies

- Boston (USA)
  - marketing and sales
- Buenos Aires (Argentina)
  - research and development

CoreLabs: the research team

Some areas of interest:

- Vulnerability research
  - Bugweek
  - Publication of advisories
- Cyber-attack planning and simulation
- Improving OS detection using neural networks

Penetration testing frameworks

Penetration testing

Actively verifying network defenses by conducting an intrusion in the same way an attacker would.

- Penetration testing tools have the ability to launch real exploits for vulnerabilities.
  - different from vulnerability scanners (Nessus, Retina, ...)
- Main tools available:
  - Core Impact (since 2001)
  - Immunity Canvas (since 2002)
  - Metasploit (since 2003)
    - open source, owned by Rapid7 since 2009

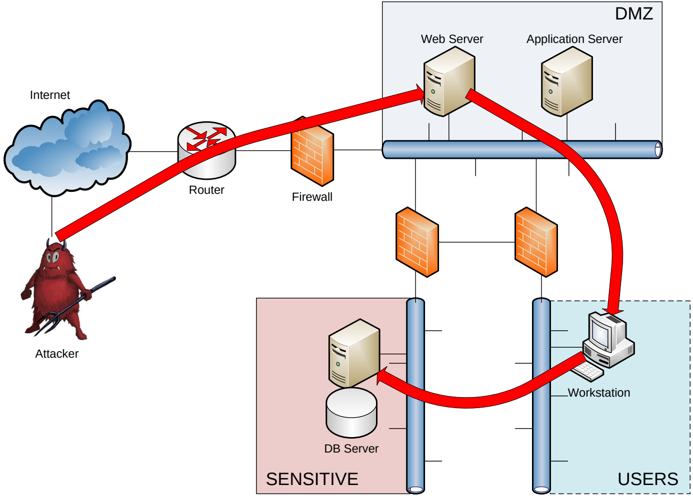
Need for automation

- Control the increasing complexity of penetration testing tools.
  - shipping more exploits
  - covering new attack vectors (Client-Side, WiFi, WebApps, ...)
- Incorporate expert knowledge to the penetration testing framework.
- Construct attack plans that pivot.

Pivoting

Compromising an intermediate machine in order to gather information or to perform attacks from that machine.

Anatomy of a real-world attack



A model for cyber-attacks

Objective of the model

- Formal representation of an attack.
- Abstraction of the penetration testing practice.
- Accurate from the attacker's point of view.

The attacker's point of view

- The attacker's main liability is the absence of knowledge about the network she wants to intrude.
- The acquisition of knowledge is an integral part of the attack.

Components of the attack model

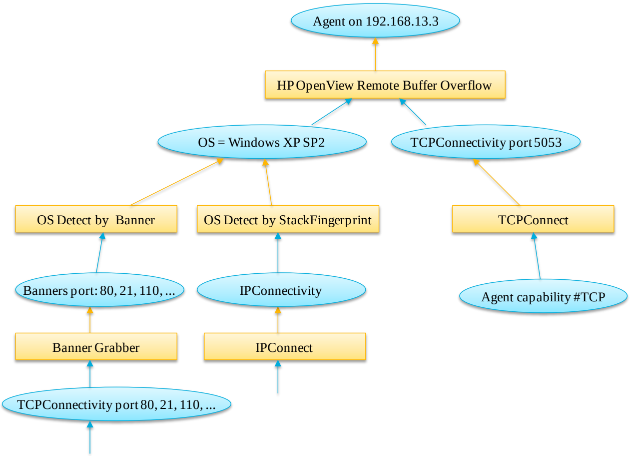
Goals
Objectives of the attack.

Assets
Anything an attacker may need during the attack.

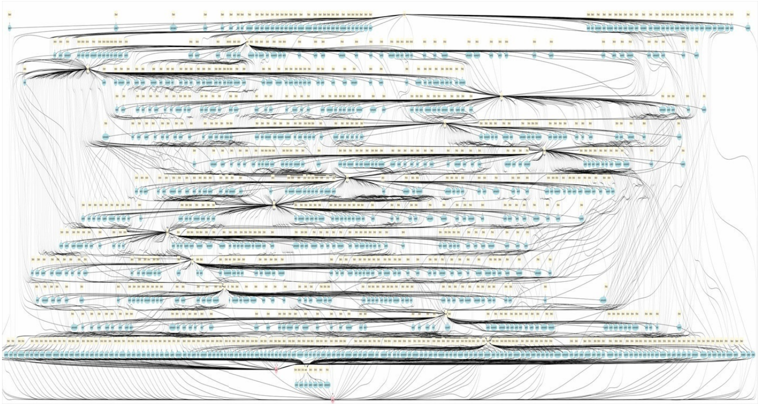
Actions
Actions are the building blocks of the attacks. They allow the obtention of assets.

Agents
Agents, whether human or software, perform the actions of the attack.

Sample attack graph

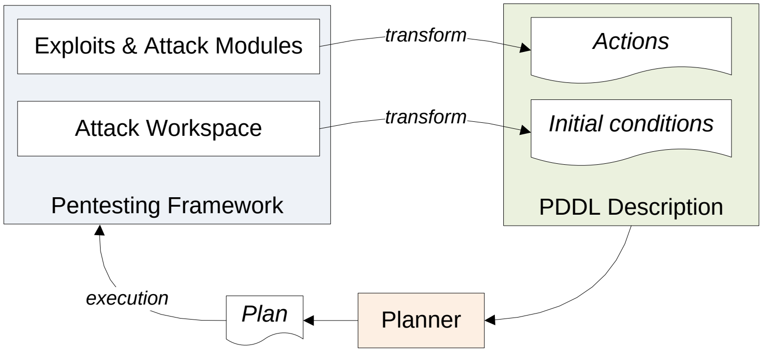


Sample attack graph (II)



From Noel and Jajodia: “Managing Attack Graph Complexity Through Visual Hierarchical Aggregation”

Architecture of our solution



Predicates for connectivity

- Assets are translated as predicates.
- Examples:
  - (connected_to_network ?s - host ?n - network)
  - (IP_connectivity ?s - host ?t - host)
  - (TCP_connectivity ?s - host ?t - host ?p - port)
  - (UDP_connectivity ?s - host ?t - host ?p - port)
- Maximum arity is 3.

Predicates for Operating System information

- Many predicates for OS information.
  - We need detailed information to evaluate the reliability of exploits.
- Examples:
  - (has_OS ?h - host ?os - operating_system)
  - (has_OS_version ?h - host ?osv - OS_version)
  - (has_OS_edition ?h - host ?ose - OS_edition)
  - (has_OS_build ?h - host ?osb - OS_build)
  - (has_OS_servicepack ?h - host ?ossp - OS_servicepack)
  - (has_architecture ?h - host ?a - OS_architecture)

Model-related actions

```
(:action TCP_connect
:parameters (?s - host ?t - host ?p - port)

:precondition (
  and (compromised ?s)
     (IP_connectivity ?s ?t)
     (TCP_listen_port ?t ?p))

:effect
  (TCP_connectivity ?s ?t ?p)
)
```

Sample exploit

```
(:action EXPLOIT_MSRPC_Samba_Command_Injection_exploit
:parameters (?s - host ?t - host)

:precondition (and
    (compromised ?s)
    (and (has_OS ?t Linux)
        (has_OS_distro ?t Ubuntu)
        (has_OS_version ?t V_6_06)
        (has_architecture ?t I386))
    (or (TCP_connectivity ?s ?t port139)
        (TCP_connectivity ?s ?t port445))
)

:effect(and
    (increase (time) 31)
    (installed_agent ?t low_privileges)
))
```

Generating test scenarios

Metrics

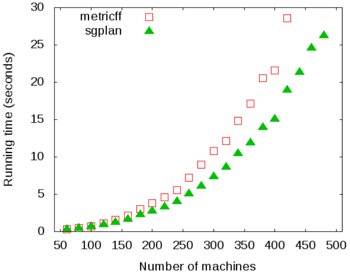
- Number of machines: up to 500
- Number of pivoting steps: up to 20
- Number of PDDL actions (exploits): up to 1800
- Number of individual predicates in the goal: up to 100

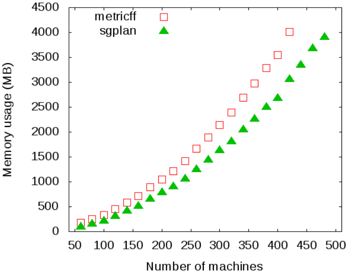
Planners

- Metric-FF (with modifications)
- SGPlan

The domain files have up to 28,000 lines.

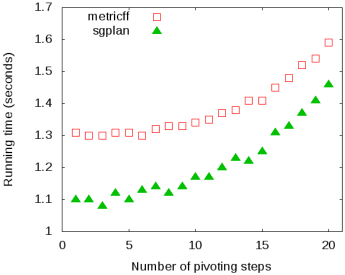
Increasing number of machines

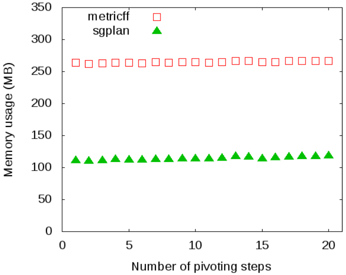




- Fixed values: 1600 actions, 1 pivoting step.
- 22 seconds, 3.2 GB of RAM to solve a 450-machine scenario with SGPlan.

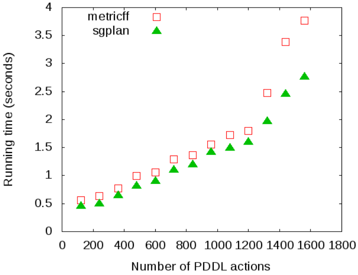
Increasing number of pivoting steps

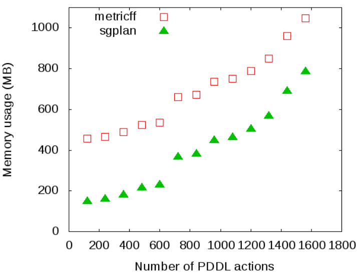




- Fixed values: 1600 actions, 120 machines.
- 1.45 seconds, 100 MB of RAM to solve a 20-step scenario with SGPlan.

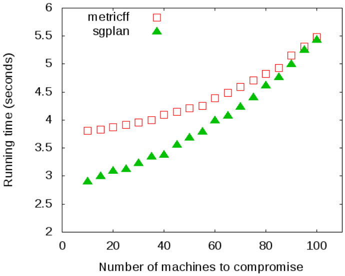
Increasing number of actions

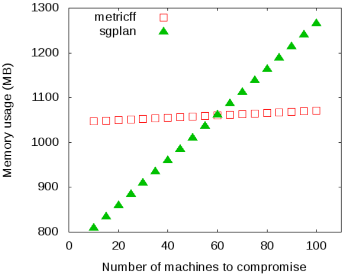




- Fixed values: 200 machines, 1 pivoting step.
- 2.75 seconds, 800 MB of RAM to solve a 1600-action scenario with SGPlan.

Increasing number of predicates in the goal





- Fixed values: 200 machines, 1 pivoting step for each compromised machine, 1600 actions.
- 5.5 seconds, 1075 MB of RAM to solve a 100-goal scenario with Metric-FF.

Demo

Demo

Summary

We have presented:

- An attack model accurate from the attacker's point of view.
- A translation of this model to PDDL.
- An implementation that uses this PDDL representation to integrate a planner to a penetration testing framework.
- An evaluation of our implementation that shows the feasability of planning and verifying attacks in real-life scenarios.

Contact information

Contact

- Jorge Lucángeli Obes: *jota@coresecurity.com*
- Carlos Sarraute: *carlos@coresecurity.com*

Email us if you would like a copy of the PDDL files.

More information

- `http://corelabs.coresecurity.com`