An Application Footprint Reference Set:

Tracking the Lifetime of Software

Mary Laamanen & John Tebbutt
National Software Reference Library
National Institute of Standards and Technology

Motivation

Gather data on the specific effects of individual software packages on a system over the software's lifetime.

Provide digital forensic investigators with new reference data.

Extend the NSRL research environment for use by forensic researchers to develop new tools and techniques.

System and Software

All software is part of the NSRL collection.
- Provides Traceability

Operating Systems

- Starting with 5 version of Microsoft operating systems.
  (XP, Vista32, Vista64, Windows7_32, Windows7_64)

Applications are chosen from the NSRL library.

Question:

What changes occur in a system when a piece of software is

- Installed?

- Executed?

- Uninstalled/Deleted?

Application Footprint

We can measure the what, where, when and how:

- Nature of changes

- Location of changes

- Stage in application “life cycle”

- Actions causing changes

Nature of Changes

Filesystem (file hashes, MAC times, etc)

- Executables
- Libraries
- Documents/Images/Multimedia
- etc.

Configuration information

- Windows Registry

Memory mapping information

- System RAM

Stage in Software Lifecycle

Depends on the package. At least:

Installation
Execution
Post-execution
Uninstallation
Post-uninstallation

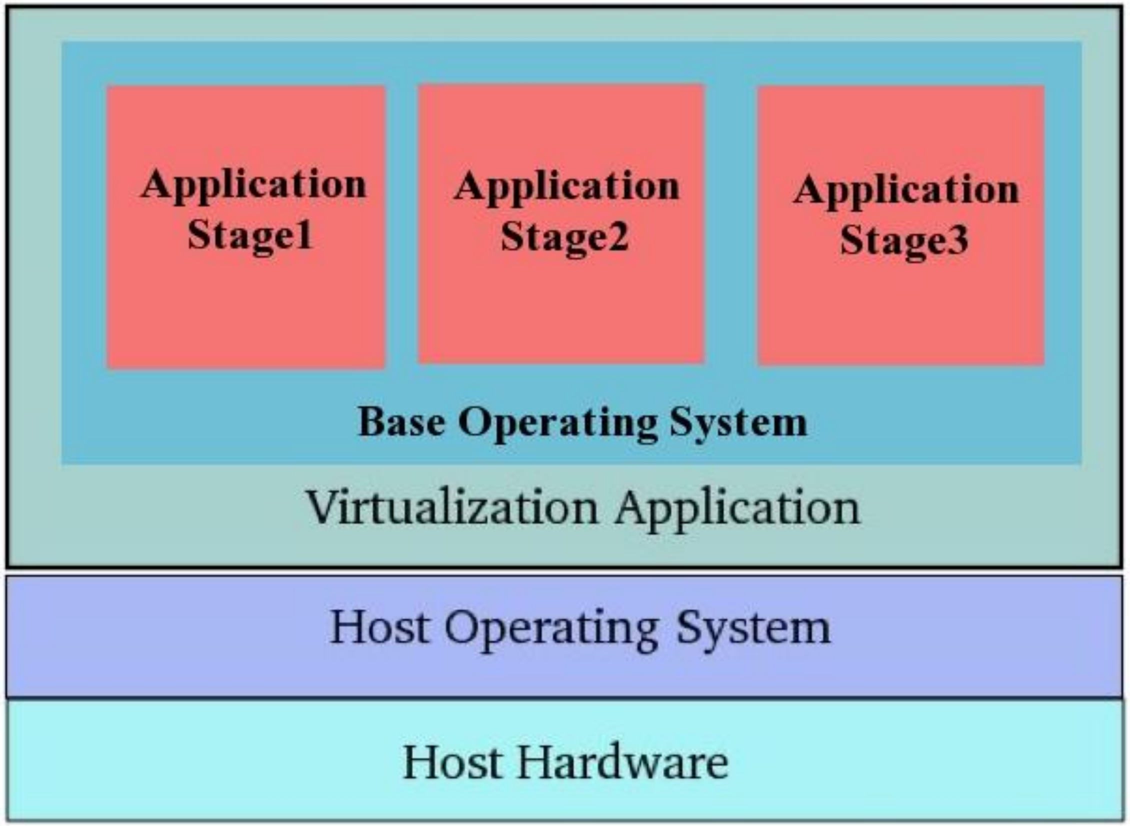
Actions Causing Changes

Particular actions during software execution may result in specific changes

e.g. visiting a web page in a browser will almost certainly add elements of the page to the browser cache. However there may be other less obvious changes...

Method

Virtual Machine Installation



Advantages

VM state can be captured at any time
- VM may be “paused” or “suspended”

VM is “frozen” as a set of files
- Hard drive, RAM contents, etc

Can be copied off for external processing...

...and saved for future reference

Application Footprint Slices

Suspend VM after each action to record the action's effects.

Capture the lifecycle of an application as
a series of suspended VMs, copied off and saved

Application Footprint is the sequence of slices derived from the stored Vm's

A “slice” contains a collection of metadata computed from a suspended VM
- file hashes, registry dumps, RAM contents
- network capture
- etc

Capturing Application Footprints

Default set of slices for each Footprint is:

After installation

After activation/registration

During execution

-The application is started, left for a short time, and the slice taken

After execution

- Close the application

After uninstallation

After restarting the Operating System

- to capture any housekeeping artifacts

How Do We Do It?

Developed tools for this process.

Need to record:
- Unique identifier for the slice
- Information about the application's state at the time the slice is generated
- All user actions when working with the application
- Unexpected behavior

Example

For each software package:

Retrieve a baseline VM image with the operating system.
Install the package.
   Save VM
Launch the software. Wait a short time.
   Save VM
Quit software.
   Save VM.
Uninstall s/w.
   Save VM
Shutdown/restart OS.
   Save VM

Application Footprint Data

NSRL data on the footprint package

- name, version, manufacturer, etc.
- date/time stamp information of the Footprint's creation (installation, execution, etc.)

Virtual machine metadata

- VM software name and version

Application Footprint Data, contd.

Operating System data:

- operating system name/version/patch level
- hardware information

Description of each slice, and the stage in the software's life cycle that it represents

Sequence of slices recording the application lifecycle

Application Footprints

Have created 35 application footprints.

Generated a total of 195 slices.

Future Plans

Process the application footprints and publish findings as part of the NSRL RDS.

- Use the current RDS format.

Generate Digital Forensics XML for artifacts of this effort.

Digital Forensics XML

DFXML provides an XML representation for a wide range of forensic information and forensic processing results.

DFXML will allow for the sharing of structured data between different forensic tools

Digital Forensics XML

NIST worked with Simson Garfinkel
Naval Postgraduate School

Extended the DFXML Schema/DTD

DFXML is part of CybOX (Cyber Observable Expression)

- http://cybox.mitre.org/

Digital Forensics XML

Interested in working with the standard and promoting it's adoption.

NIST provides a mailing list to promote discussion on this topic.

  - dfxml@nist.gov

Thank You

Mary Laamanen & John Tebbutt
National Software Reference Library
NIST
Gaithersburg, MD 20899
mary.laamanen@nist.gov
tebbutt@nist.gov