A Distributed Platform
for Network Security Assessment
Who are we?

Philippe Mathieu-Daudé  pmd@netifera.com
Claudio Castiglia       claudio@netifera.com

http://netifera.com
Agenda

- Network Security Tools
- Some Limitations of Current Tools
- The Netifera Platform
- Netifera based Applications
- Demo
- The Netifera Architecture
- Netifera Peludo
- Questions
Network Security Tools

What is a network security tool?

- Network mapping and inventory
- Port scanning and service discovery
- Vulnerability scanning
- Vulnerability exploitation
- Packet sniffing and traffic monitoring
- Password recovery
- Intrusion and malware detection
- Web application testing
Some Limitations

- Lack of Integration and Interoperability
  - Ad-hoc scripts needed (ex: translate output of a tool to the input format of another)
  - Boring repetitive tasks
  - Difficult to concentrate on the specific problem

- Lack of proper docs or good reference guides

- Sometimes difficult to understand and annoying to use
The Netifera Platform

- It is a Distributed Platform
  - Extensible & Scalable
  - Tasks can be distributed and parallelized

- Integrates Information
  - A model of the network is built as information is gathered

- Gathered Data is available to the Tools
The Netifera Platform

- **Portability**
  - Runs on any supported system w/o code changes
  - Independent of the Operating System and Architecture

- Provides Common Capabilities needed by Tools
  - File system, sockets, processes, memory, packet sniffing, crafting, injection, protocol analysis, ...
The Netifera Console
The Netifera Console

- Coordination Centre for the Platform Distribution
- Centralized Data Model
- Extensive Analysis, Exploration and Visualization Capabilities
- Solid Graphical User Interface
The Netifera Probe
The Netifera Probe

- Contains the Entire Netifera Platform... without the GUI
- Is a Node in the Distributed Platform
- Local Data Model
- Autonomous (No need to be connected to the Console)
The Netifera Probe

- Gathered information is sent to the Console
  - The Console's model integrates the information coming from the probes
  - The user is able to Analyze and Explore data from all probes
  - Enables the network to be seen from different viewpoints
The Netifera Probe

- Tools and other components can be installed, upgraded and uninstalled while running, over the network, as needed

- Easy Installation
  - Upload a single executable file (admin choice)
  - Injection inside living processes as a shellcode (pentester choice)
  - Self-Contained (no external deps)
Examples

- Security Assessment
- Network Administration
  - Management of large number of systems
  - Orchestration of tasks
- Network Monitoring
  - Monitoring of Servers/Services Healthiness
  - Detection of network based attacks
  - Network Geography
- Network Research
Video

The Java Virtual Machine As Shellcode
Inside The Netifera Platform
Console Architecture

Operating System

Native Libraries (libc, threads, graphics)

Java Virtual Machine

Java Runtime

OSGi Runtime

Netifera Framework

Netifera User Interface
Probe Architecture

- Netifera Framework
- OSGi Runtime
- Java Runtime
- Java Virtual Machine
- Libraries (libc, threads)
- in-memory dynamic linker
- Operating System

Peludo
Netifera Peludo
Toolchain to generate C based Applications

- Portable
- Self-Contained (No external dependencies)
- Injectable
- Small
PLD File Format

- It is a simple TLV based binary format

- Composed by Sections
  - Standard .code, .data, .export, .import, .reloc
  - .nimport (Native Imports)
  - Supports compression
What is an Application?

- Applications are composed by
  - Main executable
  - Dependencies (libraries)
  - Optional data files

- Under Peludo, Applications are normalized to the PLD format
  - Entirely composed of PLD files
  - Data files are embedded inside pure .data PLDs
PLD Normalization

Application

Main.exe
Library1.dll
Library2.dll
Data.dat

Normalized Application

Main.wld
Library1.wld
Library2.wld
Data.wld
A PLD Chain is a concatenation of PLD files in dependency order.
Peludo provides two components:
  - Bootstrap code
  - Peludo kernel (mainly composed by the PLD loader)
    - The PLD loader is an in-memory linker that never touches the filesystem
  - A PLB File is created when a PLD Chain is concatenated to these two components
Peludo provides two components:
- Bootstrap code
- Peludo kernel (mainly composed by the PLD loader)
  - The PLD loader is an in-memory linker that never touches the filesystem

A PLB File is created when a PLD Chain is concatenated to these two components

To launch the Application you just jump to the PLB's first byte
Injection

- In order to inject inside a living process
  - The process should be exploited (or specially created) to receive a PLB File as shellcode
  - The PLB is received and loaded into memory
  - Executed jumping to its first byte
Netifera's PLB Probe

Boot
Peludo Kernel
Peludo libraries
Native Java Runtime
Java Runtime
Knoplerfish OSGi
Netifera Framework
JamVM
Thank you!

Questions?
Contact Us

http://netifera.com
info@netifera.com