Profiling Memory Usage Patterns for Keylogging Detection with KLIMAX

Stefano Ortolani, Cristiano Giuffrida, and Bruno Crispo
{ortolani,giuffrida}@cs.vu.nl, crispo@disi.unitn.it
Vrije Universiteit, Amsterdam, The Netherlands

Research Summary

Problem: Privacy-breaching malware is designed to harvest and leak users' private data. Keylogging is the de-facto main weapon used for harvesting.

Motivation: Signature-based approaches can not keep up with the plethora of 0-day malware. State-of-the-art behavioral models are coarse-grained and massively prone to false positives.

Our approach: Taunt the malware with artificially forged keystrokes. Observe and keep track of the memory write patterns the process exhibits. Test the correlation between the pattern of injection and the memory writes pattern. Flag the process as malicious if high correlation is found.

Challenges

Memory management is a concerted process performed by the OS and the HW. Trivial approaches are a no-go. Multiple concurrent activities, of which keylogging is just one. Fine-grained approach is required. Noise produced by possible data transformations. The correlation metric must be resilient to those.

Architecture

- Three-tier architecture.
- OS independent design.
- Limited TCB.
- On-line deployment.
- On-line detection.

KLIMAX

After Instruction Execution

Windows Kernel (Ring 0)

Page Tables

Page Fault Handler

3 - Restore PTE

Classifier

4 - Update Counters

5 - Forward INT 0E

IDT

KLIMAX

Before Instruction Execution

Windows Kernel (Ring 0)

Page Tables

Page Fault Handler

3 - Override PTE

Classifier

4 - Shadow Query

5 - INT 01

IDT

KLIMAX

Injection

Monitor

Detection

KLIMAX - Kernel Level Infrastructure for Memory and eXecution profiling