Gecko: Kernel Object Mapping with Semantic Classification of Memory

Mirza Basim Baig*, Dongli Zhang*, Radu Sion†
Computer Science, Stony Brook University
{mbaig, dozhang, sion}@cs.stonybrook.edu

*Student † Faculty

Live memory forensic analysis plays a key role within the field of digital forensics and has become pivotal due to the importance of data found exclusively within running operating systems such as encryption keys, currently running processes and in-memory malware. An increasing number of applications such as kernel integrity checking, memory image forensics and virtual machine introspection rely on the complete and accurate reconstruction of this volatile data from runtime information (memory snapshot of kernel pages). However, operating systems are highly complicated and house thousands of different data structure definitions within them with possibly varied and multiple instances being present in memory at any given time. Existing kernel object mapping solutions [1] tend to work offline with a memory snapshot and try to use value or structure invariants to map out kernel memory. This severely limits the possibility to run the system in an online fashion to provide real-time protection. There is a distinct lack of behavioral/semantic analysis in the solution space of kernel object mapping. The *way* a data structure is used does not factor into the object mapping process. The real-time usage of a memory location provides additional semantic information on what type of data structure resides there that is independent of existing analysis.

Here we present the architecture for Gecko: Gecko is a hypervisor level system that enables *semantic classification* for the kernel memory space. Gecko triggers choice hardware events e.g. interrupts, packets, I/O requests etc. and observes the kernel behavior in its response to map out the semantic structure of kernel memory. Gecko is powered by the following key insight: even if a user is running a fully malicious kernel (she cannot guarantee any data or code integrity), the kernel needs to provide some basic house-keeping. For example, scheduling still needs to be done in order to run programs with some semblance of order. The kernel needs to maintain a facade of running correctly if it is to remain stealthy. E.g. the set of memory locations that are used periodically in relation to the clock interrupt form a superset of those memory locations that are involved in scheduling. This leads to a semantic segregation of the memory space into different classes that represent their actual high level usage. Gecko can use this classification to bootstrap kernel object mapping and vastly narrow down search space size as compared to previous approaches [2] (that use the whole memory as the working area).

References

[1]. Martim Carbone, Weidong Cui, Long Lu, Wenke Lee, Marcus Peinado, and Xuxian Jiang. Mapping kernel objects to enable systematic integrity checking. In Proceedings of the 16th ACM conference on Computer and communications security, CCS '09, pages 555–565, New York, NY, USA, 2009. ACM.

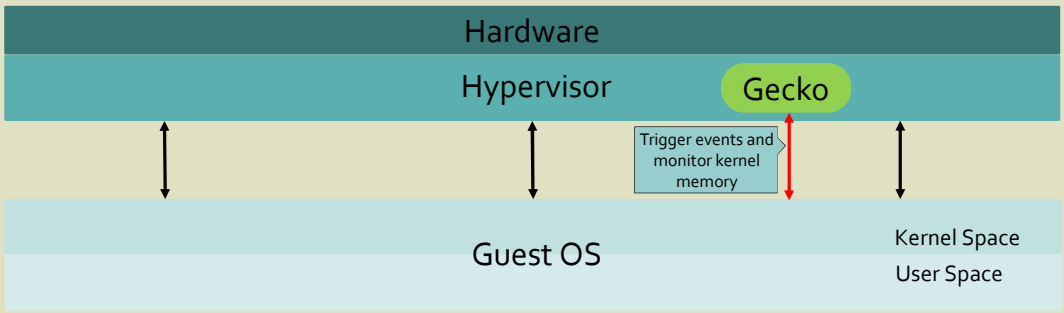
[2]. SigGraph: Brute Force Scanning of Kernel Data Structure Instances Using Graph-based Signatures. Zhiqiang Lin, Junghwan Rhee, Xiangyu Zhang, Dongyan Xu, Xuxian Jiang. NDSS2011.

Gecko – Kernel Object Mapping With Semantic Classification of Memory

Mirza Basim Baig
Dongli Zhang
Radu Sion

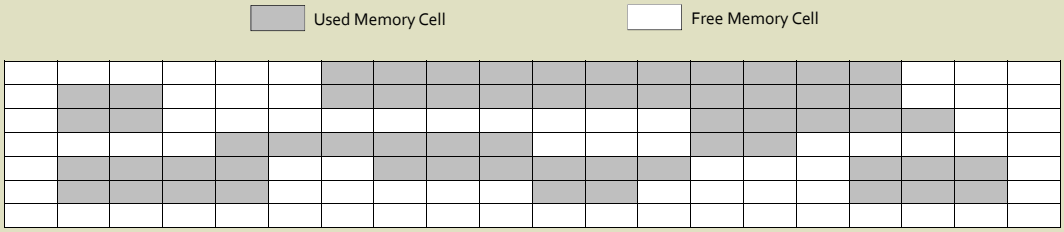
NSAC Stony Brook Network Security and Applied Cryptography Lab

The Virtualization Stack



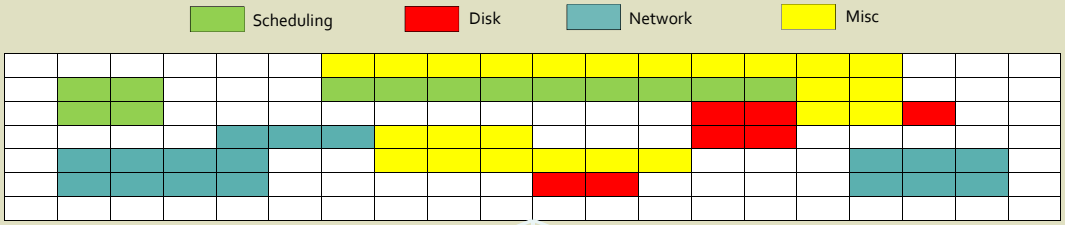
1 **Gecko**[1] sits inside the Hypervisor and triggers events of interest (clock interrupts, network packets, I/O calls)

Kernel Memory Space

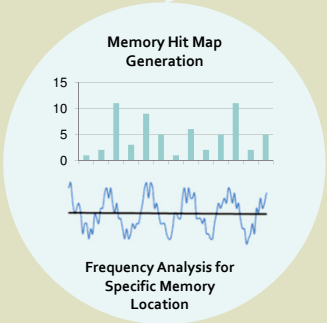


2 **Gecko** keeps track of memory touched in response to handle specific events

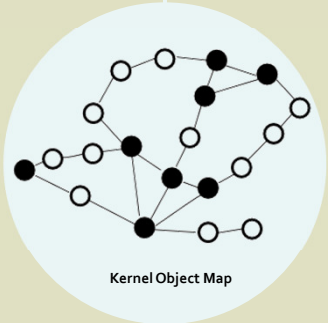
Semantically Tagged Kernel Memory Space



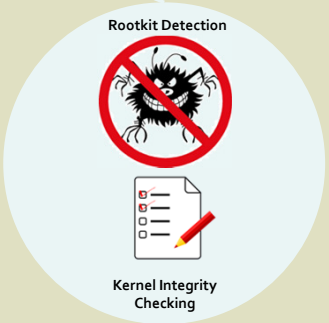
3 **Gecko** generates *semantic tags* for each used memory cell. Semantic tags describe what event memory cells are used for



Gecko builds memory access profile and infers patterns to assign semantic tags



Semantically Tagged Memory enables kernel object reconstruction



Kernel Object Maps are used to **detect rootkits** and **check kernel integrity**

4 **Semantically tagged memory** enables a variety of useful applications such as **Kernel Object Mapping, Rootkit Detection and Kernel Integrity Checking**

[1]**Gecko**: **Ge**nerating Semantic **C**lassification For **K**ernel **O**bjects

Like to chat or comment? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

Stony Brook University