
CSS
2016·第二届
中国互联网安全领袖峰会
Cyber Security Summit
智慧安全 连接赋能

Graffiti: the spraying attacks slayer

Mariano Graziano, Security researcher, Cisco Talos

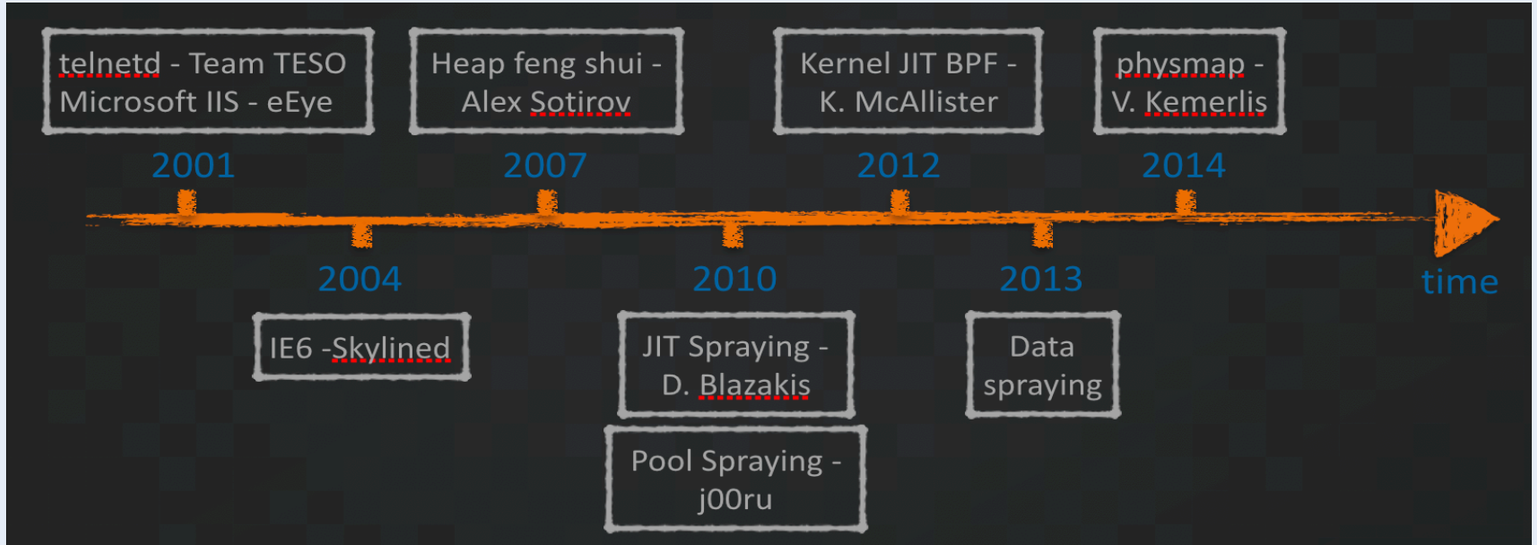
whoami

- Security researcher at Cisco Talos
- Ph.D. from Telecom ParisTech/Eurecom
- Hackademic
- Malware analysis / Memory forensics / Mitigations

SPRAYING

- Traditional code based spraying
- JIT spraying
- Data spraying and stack pivoting

SPRAYING - TIMELINE



SPRAYING – 64 bit

- Spraying still a valuable technique
  - UAF vulnerabilities
  - Flaws in ASLR implementation
  - Particular types of vulnerabilities[1]
  - 32bit processes in 64bit OS[2]

[1] https://ifsec.blogspot.it/2013/11/exploiting-internet-explorer-11-64-bit.htm
[2] http://blog.skylined.nl/20160622001.html

RELATED WORK

HEAP SPRAYING

- Egele et al. – DIMVA 09
- Nozzle – USENIX Security 09
- Bubble – ESSoS 10

RELATED WORK

HEAP SPRAYING

- Egele et al. – DIMVA 09
- Nozzle – USENIX Security 09
- Bubble – ESSoS 10

JIT SPRAYING

- JITsec – VEE 06
- Bania – Whitepaper 10
- JITDefender – IFIP 10
- Lobotomy – Ares 14

RELATED WORK

HEAP SPRAYING

- Egele et al. – DIMVA 09
- Nozzle – USENIX Security 09
- Bubble – ESSoS 10

JIT SPRAYING

- JITsec – VEE 06
- Bania – Whitepaper 10
- JITDefender – IFIP 10
- Lobotomy – Ares 14

DATA SPRAYING

- EMET – Microsoft 09
- Browser solutions

RELATED WORK

HEAP SPRAYING

- Egele et al. – DIMVA 09
- Nozzle – USENIX Security 09
- Bubble – ESSoS 10

JIT SPRAYING

- JITsec – VEE 06
- Ranjia – Whitepaper 10
- JITDefender – IFIP 10
- Lobotomy – Ares 14

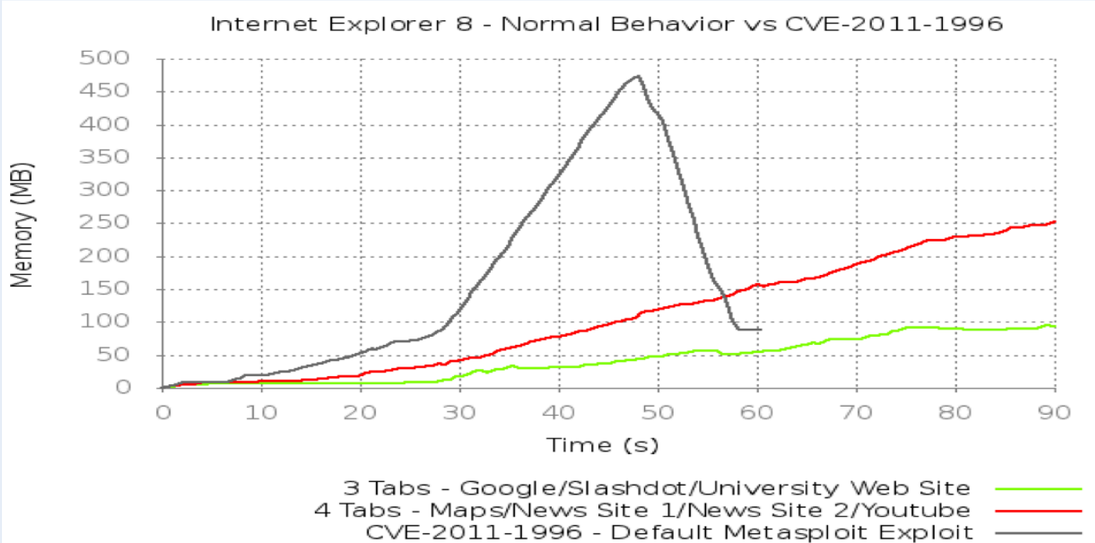
DATA SPRAYING

- EMET – Microsoft 09
- Browser solutions

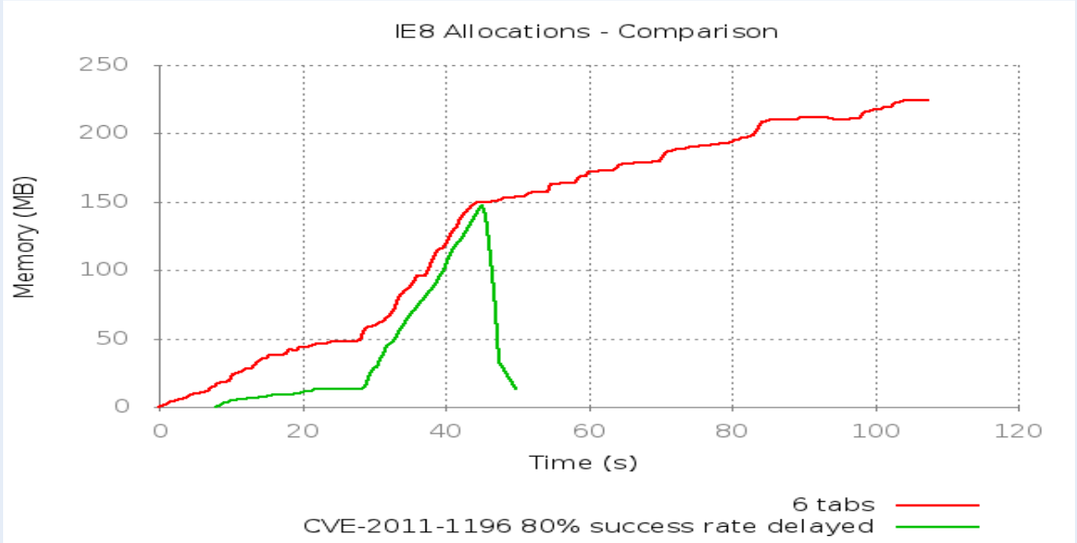
No comprehensive and scalable solutions vs spraying

No OS agnostic solutions

MEMORY FOOTPRINT



MEMORY FOOTPRINT



RELIABILITY – HEAP SPRAYING

- 80%: 131 MB – CVE-2011-1996
- 0%: < 131 MB - CVE-2011-1996
- Possible Watering hole scenarios

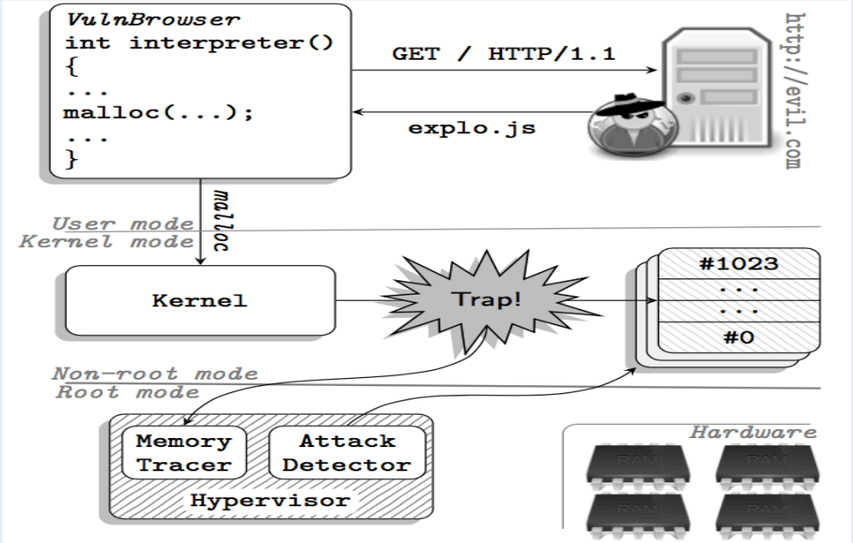
GOALS

- The system should be completely independent from the memory allocator used by the protected application
- The system should not require any OS dependent information
- The overhead introduced by the system should be “reasonable”
- Modular framework based on plugins

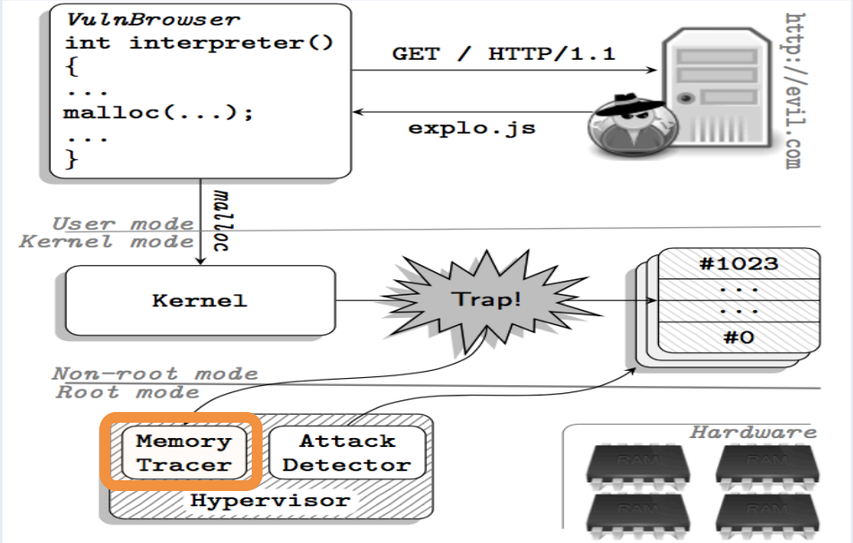
VIRTUALIZATION – VT-x

- `vmexit/vmentry` transitions
- `vmcall` instruction
- Virtual Machine Control Structure (`VMCS`)
- Extended Page Tables (`EPT`):
  - EPT misconfigurations
  - EPT violations

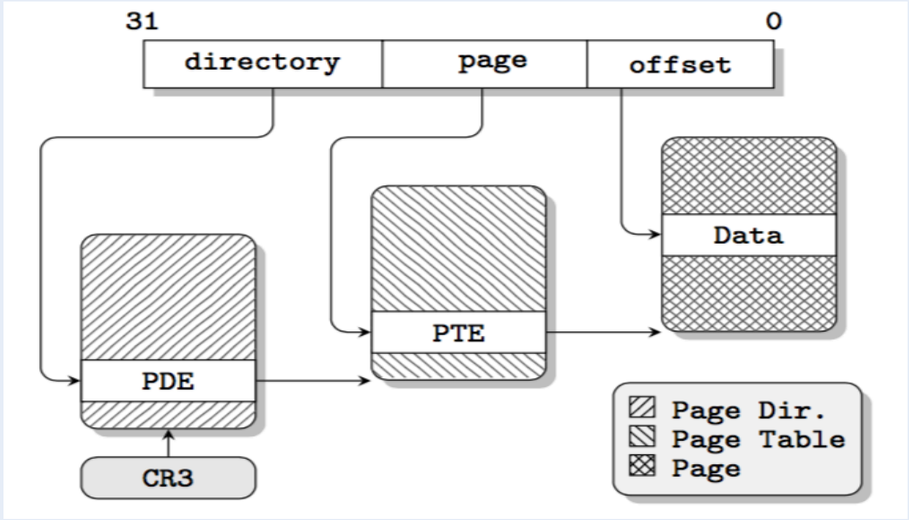
ARCHITECTURE OVERVIEW



ARCHITECTURE OVERVIEW



MEMORY TRACER



MEMORY TRACER

- Detection by looking at the PTEs:
  - page creation/modification/removal
- Detection by looking at the PDEs:
  - page table creation/modification/removal

INTERFERENCE PROBLEM

- Overhead issues:
  - A modification of the memory page of the running process creates a side effect modification of a memory page in another process
  - Due to kernel memory optimizations

INTERFERENCE PROBLEM

- Overhead for Internet Explorer 10 (IE): 22%
- Overhead for Acrobat Reader and Firefox on top of IE10: 63%
- IE's overhead increased from 22% to 63%

MICRO-VIRTUALIZATION

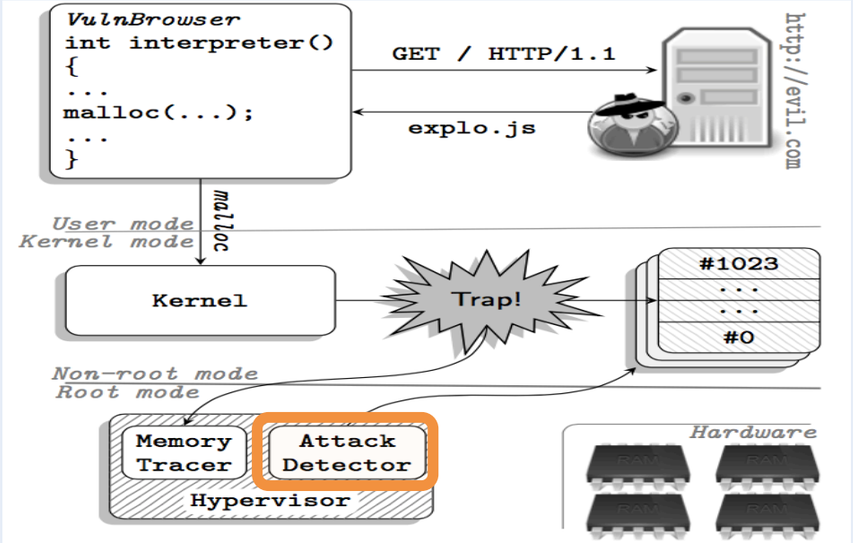
- Each monitored process runs inside its own virtual memory sandbox
- Graffiti enables the memory protection only for actual running and monitored process

MICRO-VIRTUALIZATION - DETAILS

- Micro-virtualization is based on EPT:
  - Select a set of physical pages to monitor the target process
  - One EPT Pointer (EPTP) per process

DEMO 0x00

ARCHITECTURE OVERVIEW



STATIC ANALYZER

- Set of heuristics to detect the different spraying techniques:
  - Malicious code detector [ACSAC10, SEC11]

STATIC ANALYZER

- Set of heuristics to detect the different spraying techniques:
  - Malicious code detector [ACSAC10, SEC11]
  - Self-unpacking shellcode detector [ACSAC07]

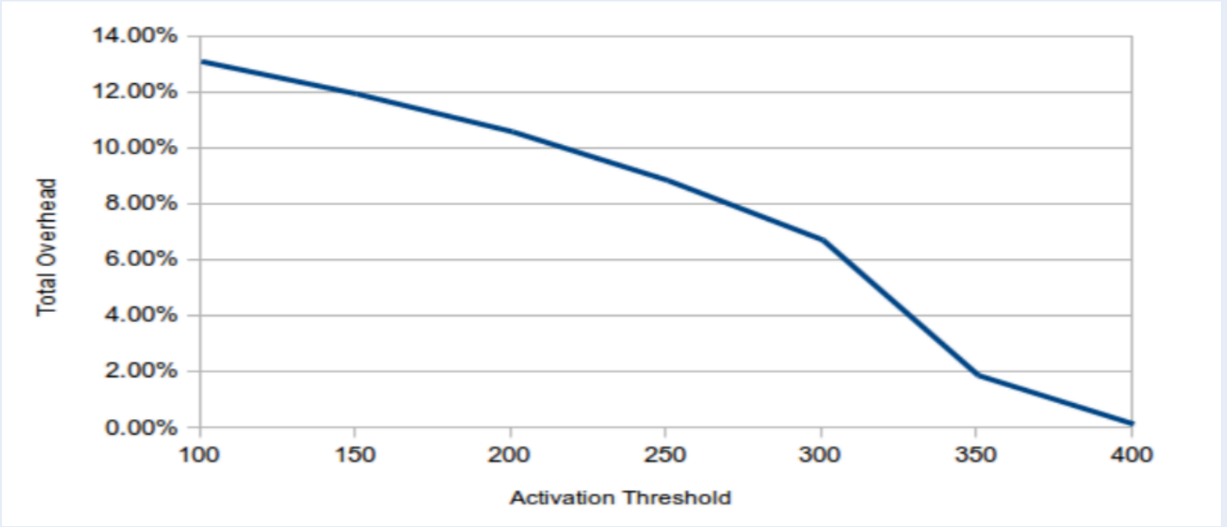
STATIC ANALYZER

- Set of heuristics to detect the different spraying techniques:
  - Malicious code detector [ACSAC10, SEC11]
  - Self-unpacking shellcode detector [ACSAC07]
  - Data spraying detector [RAID13, RAID15]

EXPERIMENTS: OVERHEAD

- The memory tracer is always active:
- Stress suite results (8MB every 2s):
  - Windows 7: 24%
  - Linux 3.2: 25%

EXPERIMENTS: OVERHEAD



DETECTION ACCURACY

CVE	Application	Exploit Technique	Detected
2010-0248	Adobe Flash player	ROP + packed sc	Yes
2011-0609	Adobe Reader	JIT + packed sc	Yes
2011-2462	Adobe Reader	ROP + packed sc	Yes
2010-2883	Adobe Reader	Ret2Lib + packed sc	Yes
2011-1996	IExplorer	ROP	Yes
2009-2477	Firefox	Plain Shellcode	Yes

DETECTION ACCURACY

- System tested with 1000 malicious PDF, 1000 benign PDF documents and top 1000 Alexa websites.
- Conservative threshold (150MB)
- Graffiti detected all the attacks with zero false alarms

GLOBAL EXPERIMENTS

- Final global test in which real users use Graffiti during their everyday activities for a total of 8-10 hours per day in a period of 7 days.
- Conservative threshold of 150MB on IE8 for Windows 7 machines

GLOBAL EXPERIMENTS

- 492 distinct web pages visited
- Detector activated 55 times (~8 times per day)
- 12 alerts on pages that seemed to be benign. A closer inspection of the false positive shows the data spraying to be the only responsible

GRAFFITI - CONFIGURATION

- godfather/hyperdbg/hyperdbg_host.c:
  - #define TARGET1 “iexplorer.exe”
- godfather/hyperdbg/godfather.c:
  - #define EXPLOIT_THRESHOLD THRESHOLD_150MB
- Compilation:
  - make -f Makefile.windows

DEMO

DEMO 0x01

LIMITATIONS

- Possible evasion of the three heuristics described:
  - Code pointer frequency
  - Shellcode frequency analysis
- Big chunks and then ad-hoc allocation functions
- Current supported OS: Windows and Linux
- Architecture dependent (Intel)

FUTURE

- Propose stronger heuristics
- More comprehensive testing:
  - New browser versions
  - Kernel Heap spraying

CONCLUSIONS

- First efficient and comprehensive solution to defeat spraying
- Micro-virtualization
- Open source

SOURCE CODE & PAPER

- https://github.com/graffiti-hypervisor/graffiti-hypervisor

- “Micro-Virtualization Memory Tracing to Detect and Prevent Spraying Attacks”

Stefano Cristalli, Mattia Pagnozzi, Mariano Graziano, Andrea Lanzi, Davide Balzarotti

25th USENIX Security Symposium

2016·第二届
中国互联网安全领袖峰会
Cyber Security Summit

THANK YOU

智慧安全　连接赋能

magrazia@cisco.com
@emd3l