European Commission
Directorate-General Home Affairs

Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme

HOME/2009/CIPS/AG/C2-050
i-Code: Real-time Malicious Code Identification

Deliverable D4.1: Midterm Dissemination Report

Workpackage:	WP4: <Dissemination>
Contractual delivery date:	June 2011
Actual delivery date:	July 2011
Deliverable Dissemination Level:	Public
Editor	Stefano Zanero (POLIMI)
Contributors	all partners
Internal Reviewers:	Michalis Polychronakis

Executive Summary: This is the first year dissemination report for the i-Code project.

With the support of the Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme. European Commission - Directorate-General Home Affairs[†].

[†]This project has been funded with the support of the Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme of European Commission - Directorate-General Home Affairs. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Contents

1 Introduction

This report summarizes the dissemination activities carried out by the *i-Code* project in its first year. We will list the papers presented by the *i-Code* consortium in international, peer reviewed conferences.

During the period covered by this report, the *i-Code* consortium produced a total of **6 conference papers**.

2 Papers presented at International Conferences

The following papers were accepted and presented at international, peer-reviewed conferences during the first twelve months of the project:

[1] Zacharias Tzermias, Giorgos Sykiotakis, Michalis Polychronakis, and Evangelos P. Markatos. Combining static and dynamic analysis for the detection of malicious documents. In *Proceedings of the European Workshop on System Security (EuroSec)*, April 2011.

[2] Michalis Polychronakis, Kostas G. Anagnostakis, and Evangelos P. Markatos. Comprehensive shellcode detection using runtime heuristics. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC)*, December 2010.

[3] Giorgos Vasiliadis, Michalis Polychronakis, and Sotiris Ioannidis. GPU-assisted malware. In *Proceedings of the 5th IEEE International Conference on Malicious and Unwanted Software (MALWARE)*, October 2010.

[4] Andrea Lanzi, Davide Balzarotti, Christopher Kruegel, Mihai Christodorescu, and Engin Kirda. Accessminer: using system-centric models for malware protection. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 399–412, New York, NY, USA, 2010. ACM.

[5] Asia Slowinska, Traian Stancescu, and Herbert Bos. Howard: a dynamic excavator for reverse engineering data structures. In *Proceedings of NDSS 2011*, San Diego, CA, 2011.

[6] Georgios Portokalidis, Philip Homburg, Kostas Anagnostakis, and Herbert Bos. Paranoid android: Versatile protection for smartphones. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC)*, Austin, Texas, December 2010.

In [1] we present MDScan, a standalone malicious document scanner that combines static document analysis and dynamic code execution to detect previously unknown PDF threats. PDF files are nowadays a prime vector for malware propagation. As vulnerabilities in the major PDF viewers keep surfacing, effective detection of malicious PDF documents is an important issue that we try to address: our evaluation shows that MDScan can detect a broad range of malicious PDF documents, even when they have been extensively obfuscated.

In [2] we present Gene, a code injection attack detection system based on passive network monitoring. Gene is a comprehensive shellcode detection tool that uses a set of runtime heuristics to identify the presence of shellcode in arbitrary data streams. We have identified fundamental machine-level operations that are inescapably performed by different shellcode types, based on which we have designed heuristics that enable the detection of plain and metamorphic shellcode regardless of the use of self-decryption (which is the characteristic most other systems use). Our experimental evaluation and real-world deployment show that Gene can effectively detect a large and diverse set of shellcode samples that are currently missed by existing detectors, while so far it has not generated any false positives.

In [3] we design and implement unpacking and run-time polymorphism for a GPU, and tested them using existing graphics hardware. We also discuss how upcoming GPU features can be utilized to build even more robust, evasive, and functional malware. Malware writers constantly seek new methods to obfuscate their code so as to evade detection by virus scanners. Two code-armoring techniques that pose significant challenges to existing malicious-code detection and analysis systems are unpacking and run-time polymorphism. We demonstrate how malware can increase its robustness against detection by taking advantage of the ubiquitous Graphics Processing Unit.

In [5] we present a new solution, known as Howard, to extract data structures from C binaries without any need for symbol tables. Our results are significantly more accurate than those of previous methods, sufficiently so to allow us to generate our own (partial) symbol tables without access to source code. This makes debugging and analyzing such binaries simpler. Also, we show that we can protect existing binaries from popular memory corruption attacks, without access to source code. Unlike most existing tools, our system uses dynamic analysis (on a QEMU-based emulator) and detects data structures by tracking how a program uses memory.

In [6] we discuss a solution for protection of smartphones which we named Paranoid Android. Since smartphones have limited computational power and are extremely sensitive to battery consumption, we propose an alternative solution, where security checks are applied on remote security servers that host exact replicas of the phones in virtual environments. The servers are not subject to the same constraints, allowing us to apply multiple detection techniques simultaneously. We implemented a prototype of this security model for Android phones, and show that it is both practical and scalable (one server can protect hundreds of devices).

3 Presence in the media

In fall 2010, our paper on GPU malware [3] stirred quite a lot of discussion in popular online media. In addition to several popular tech-news websites that featured the paper(see Figure 1), we were also interviewed on the subject.

- Slashdot:
  htt://it.slashdot.org/story/10/09/27/1422205/Malware-Running...
- The Inquirer (Sotiris Ioannidis interview):
  http://www.theinquirer.net/inquirer/feature/1938790/sotiris...
- The Register:
  http://www.theregister.co.uk/2010/09/28/gpu_assisted_malware/
- Hot Hardware:
  http://www.hothardware.com/News/New-Whitepaper-Claims-GPUs...

4 Project website

The website of the project, featuring all the public and dissemination information, can be reached at http://www.icode-project.eu. A screenshot of a page of the website, featuring all of the project publications, can be seen in Figure 2.

It has been operational since early August 2010. People from more than 60 countries have accessed the web site. Most of the accesses are from Europe and the United States of America. A complete set of statistics can be seen in Figure 3.

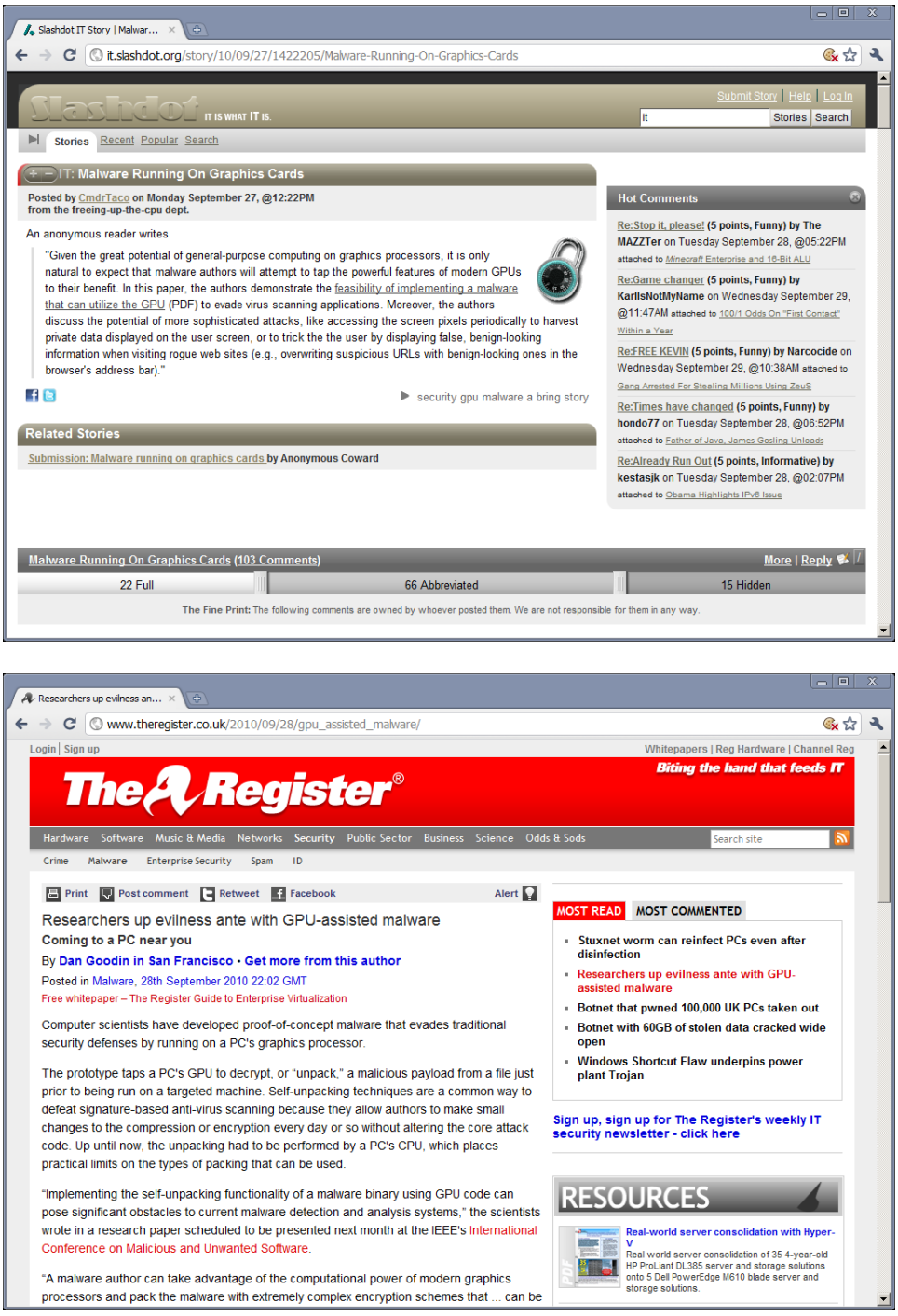

Figure 1: Our GPU malware paper [3] as featured in populare tech-news websites.

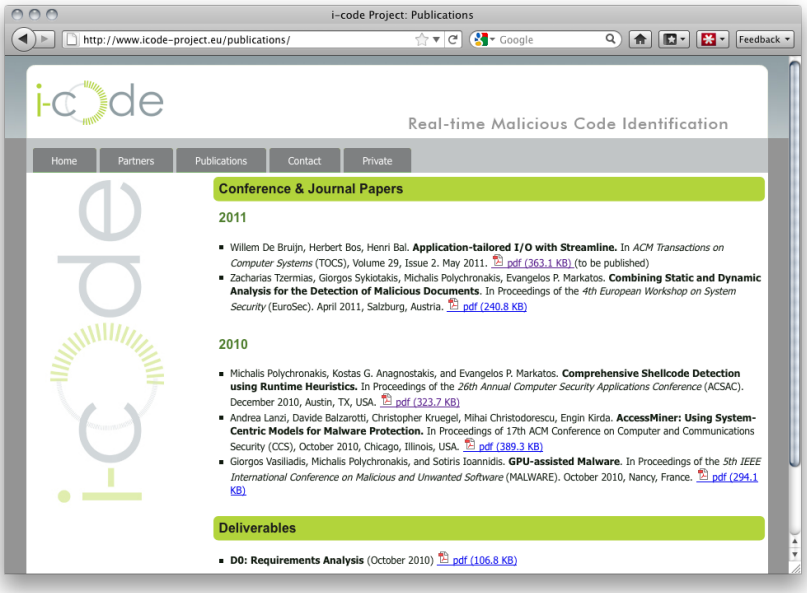

Figure 2: A screenshot of a page of the project website

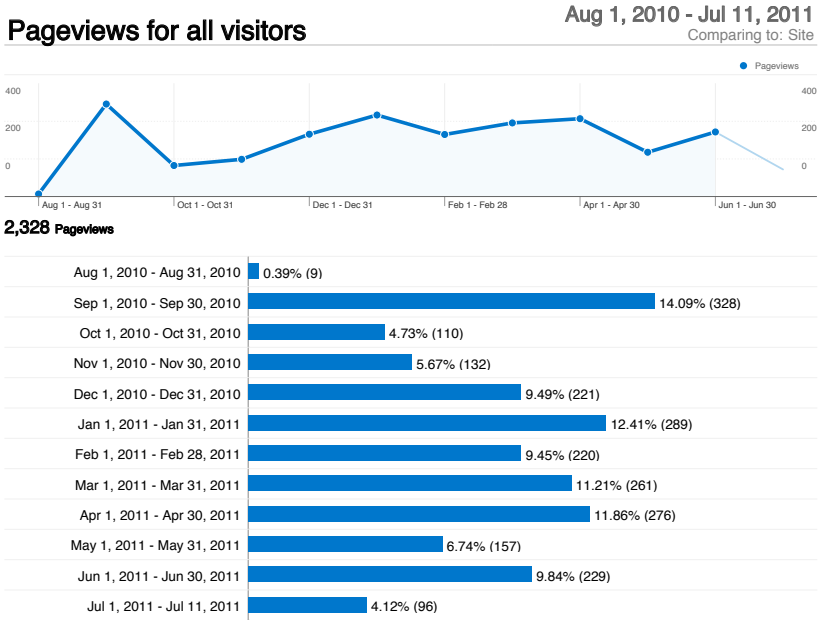

Figure 3: The complete access statistics of the project website