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 D4.2: Final Dissemination Report

Workpackage:	WP4: Dissemination
Contractual delivery date:	June 2012
Actual delivery date:	July 2012
Deliverable Dissemination Level:	Public
Editor	Stefano Zanero (POLIMI)
Contributors	all partners
Internal Reviewers:	Michalis Polychronakis (FORTH)

Executive Summary: This is the dissemination report for the second year of 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 during its second and final year. We list the papers presented by the *i-Code* consortium in international, peer reviewed conferences, discuss the presence of the project in various media, provide visitor statistics of the project website, and provide a summary of the activities from the final i-Code conference.

2 Papers presented at International Conferences

The following papers were accepted and presented at international, peer-reviewed conferences during the second year of the project:

[1] Davide Canali, Andrea Lanzi, Davide Balzarotti, Mihai Christoderescu, Christopher Kruegel, and Engin Kirda. A quantitative study of accuracy in system call-based malware detection. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*, July 2012.

[2] Andrei Bacs, Remco Vermeulen, Asia Slowinska, and Herbert Bos. System-level support for intrusion recovery. In *DIMVA'12*, Heraklion, Greece, July 2012.

[3] Christian Rossow, Christian J. Dietrich, Chris Grier, Christian Kreibich, Vern Paxson, Norbert Pohlmann, Herbert Bos, and Maarten van Steen. Prudent Practices for Designing Malware Experiments: Status Quo and Outlook. In *Proceedings of IEEE Security & Privacy (Oakland) 2012*, May 2012.

[4] Matthias Neugschwandtner, Paolo Milani Comparetti, and Christian Platzer. Detecting Malware's Failover C&C Strategies with SQUEEZE. In *Proceedings of ACSAC'11*, Orlando, FL, December 2011.

[5] Matthias Neugschwandtner, Paolo Milani Comparetti, Gregoire Jacob, and Christopher Kruegel. ForeCast - Skimming off the Malware Cream. In *Proceedings of ACSAC'11*, Orlando, FL, December 2011.

[6] Giorgos Vasiliadis, Michalis Polychronakis, and Sotiris Ioannidis. Parallelization and Characterization of Pattern Matching using GPUs. In *Proceedings of the 2011 IEEE International Symposium on Workload Characterization (IISWC)*, Austin, TX, USA, November 2011.

[7] Giorgos Vasiliadis, Michalis Polychronakis, and Sotiris Ioannidis. MIDeA: A Multi-Parallel Intrusion Detection Architecture. In *Proceedings of the 18th ACM/SIGSAC Conference on Computer and Communications Security (CCS)*, Chicago, IL, USA, October 2011.

[8] Erik Bosman, Asia Slowinska, and Herbert Bos. Minemu: The world's fastest taint tracker. In *Proceedings of RAID'11*, Menlo Park, CA, September 2011.

[9] Martina Lindorfer, Clemens Kolbitsch, and Paolo Milani Comparetti. Detecting Environment-Sensitive Malware. In *Proceedings of RAID'11*, Menlo Park, CA, September 2011.

[10] Martin Szydlowski, Ben Y. Zhao, Engin Kirda, and Christopher Kruegel. BTLab: A System-Centric, Data-Driven Analysis and Measurement Platform for BitTorrent Clients. In *Proceedings of ICCCN'11*, Maui, HI, August 2011.

In [1] we present a systematic approach to measure how the choice of behavioral models influences the quality of a malware detector. We tackle this problem by executing a large number of testing experiments, in which we explored the parameter space of over 200 different models, corresponding to more than 220 millions of signatures. Our results suggest that commonly held beliefs about simple models are incorrect in how they relate changes in complexity to changes in detection accuracy. This implies that accuracy is non-linear across the model space, and that analytical reasoning is insufficient for finding an optimal model, and has to be supplemented by testing and empirical measurements.

In [2], we describe DiskDuster, an Argos-based system capable of recovering after an intrusion. Recovering from attacks is hard and gets harder as the time between the initial infection and its detection increases. Which files did the attackers modify? Did any of user data depend on malicious inputs? Can I still trust my own documents or binaries? When malicious code has been active for some time and its actions are mixed with those of benign applications, these questions are impossible to answer on current systems. In this paper, we describe DiskDuster, an attack analysis and recovery system capable of recovering from complicated attacks in a semi-automated manner. DiskDuster traces malicious code at byte-level granularity both in memory and on disk in a modified version of QEMU. Using taint analysis, DiskDuster also tracks all bytes written by the malicious code, to provide a detailed view on what (bytes in) files derive from malicious data. Next, it uses this information to remove malicious actions at recovery time.

In [3], we describe guidelines for sound experimentation with malware data sets. Malware researchers rely on the observation of malicious code in execution to collect datasets for a wide array of experiments, including generation of detection models, study of longitudinal behavior, and validation of prior research. For such research to reflect prudent science, the work needs to address a number of concerns relating to the correct and representative use of the datasets, presentation of methodology in a fashion sufficiently transparent to enable reproducibility, and due consideration of the need not to harm others. In this paper we study the methodological rigor and prudence in 36 academic publications from 2006-2011 that rely on malware execution. 40% of these papers appeared in the 6 highest-ranked academic security conferences. We find frequent shortcomings, including problematic assumptions regarding the use of execution-driven datasets (25% of the papers), absence of description of security precautions taken during experiments (71% of the articles), and oftentimes insufficient description of the experimental setup.

In [4], we present Squeeze, a system to detect failover command and control (C&C) strategies in malware. The ability to remote- control infected PCs is a fundamental component of modern malware campaigns. At the same time, C&C infrastructure that provides this capability is an attractive target for mitigation. In recent years, more or less successful takedown operations have been conducted against botnets employing both client-server and peer-to-peer C&C architectures. To improve their robustness against such disruptions of their illegal business, botnet operators routinely deploy redundant C&C infrastructure and implement failover C&C strategies. Squeeze uses techniques based on multi-path exploration to discover how malware behaves when faced with the simulated take-down of some of the network endpoints it communicates with. We show that Squeeze allows us to detect backup C&C servers, increasing the coverage of an automatically generated C&C blacklist by 19.7%, and can trigger domain generation algorithms that malware implements for disaster-recovery.

In [5] we present ForeCast, a novel approach to malware sample selection that attempts to maximize the total value of the information obtained from analysis, according to an application-dependent scoring function. We leverage previous work on behavioral malware clustering and introduce a machine-learning-based system that uses all statically-available information to predict into which behavioral class a sample will fall, before the sample is actually executed. We discuss scoring functions tailored at two practical applications of large-scale dynamic analysis: the compilation of network blacklists of command and control servers and the generation of remediation procedures for malware infections. The large-scale evaluation on over 600,000 malware samples shows that our prototype can increase the amount of potential command and control servers detected by up to 137% over a random selection strategy and 54% over a selection strategy based on sample diversity.

In [6], we explore how the highly parallel computational capabilities of commodity graphics processing units (GPUs) can be exploited for high-speed pattern matching. We present the design, implementation, and evaluation of a pattern matching library running on the GPU, which can be used transparently by a wide range of applications to increase their overall performance. The library supports both string searching and regular expression matching on the NVIDIA CUDA architecture. We have also explored the performance impact of different types of memory hierarchies, and present solutions to alleviate memory congestion problems. The results of our performance evaluation using off-the-self graphics processors demonstrate that GPU-based pattern matching can reach tens of gigabits per second on different workloads.

In [7], we present a multi-parallel intrusion detection architecture tailored for high speed networks. To cope with the increased processing throughput requirements, our system parallelizes network traffic processing and analysis at three levels, using multi-queue NICs, multiple CPUs, and multiple GPUs. The proposed design avoids locking, optimizes data transfers between the different processing units, and speeds up data processing by mapping different operations to the processing units where they are best suited. Our experimental evaluation shows that our prototype implementation based on commodity off-the-shelf equipment can reach processing speeds of up to 5.2 Gbit/s with zero packet loss when analyzing traffic in a real network, whereas the pattern matching engine alone reaches speeds of up to 70 Gbit/s, which is an almost four times improvement over prior solutions that use specialized hardware.

In [8], we present Minemu—a fast intrusion detection system by means of dynamic taint analysis. Dynamic taint analysis is a very effective way of detecting and defending against memory corruption attacks (and it is also used by Argos), yet large performance penalties prevent its widespread use. Minemu incurs a slowdown of only 1.5x-3x for real-world applications. Possibly fast enough for production systems.

In [9] we describe Disarm, a system for screening malware samples for evasive behavior. As dynamic malware analysis sandboxes increase in popularity, they are faced with the problem of malicious code detecting the instrumented environment to evade analysis. In the absence of an "undetectable", fully transparent analysis sandbox, defense against sandbox evasion is mostly reactive: Sandbox developers and operators tweak their systems to thwart individual evasion techniques as they become aware of them, leading to a never-ending arms race. Disarm uses novel techniques for detecting malware samples that exhibit semantically different behavior across different analysis sandboxes. These techniques are compatible with any monitoring technology that can be used for dynamic analysis, and are completely agnostic to the way that malware achieves evasion. We demonstrate that Disarm can accurately detect evasive malware, leading to the discovery of previously unknown evasion techniques.

In [10] we present BTLab, a distributed platform to measure and analyze the differences between BitTorrent clients. Due to extensibility, and a certain vagueness in the BitTorrent specification, many clients diverge in some aspects from each other. Most research to date disregarded the effects of these differences. BTLab allows us to create and control BitTorrent swarms, composed of hundreds of clients of our choice. We use captured network traffic to measure the performance and uncover the reasons for observed differences. For our experiments, we deployed BTLab on a cluster and on Planetlab and selected four popular BitTorrent clients. Our analysis reveals flaws in piece selection and connection management algorithms that adversely affect the performance of some clients.

3 Presence in the media

During the second year of the project, members of the project have discussed i-Code in various articles, radio broadcasts, and television shows.

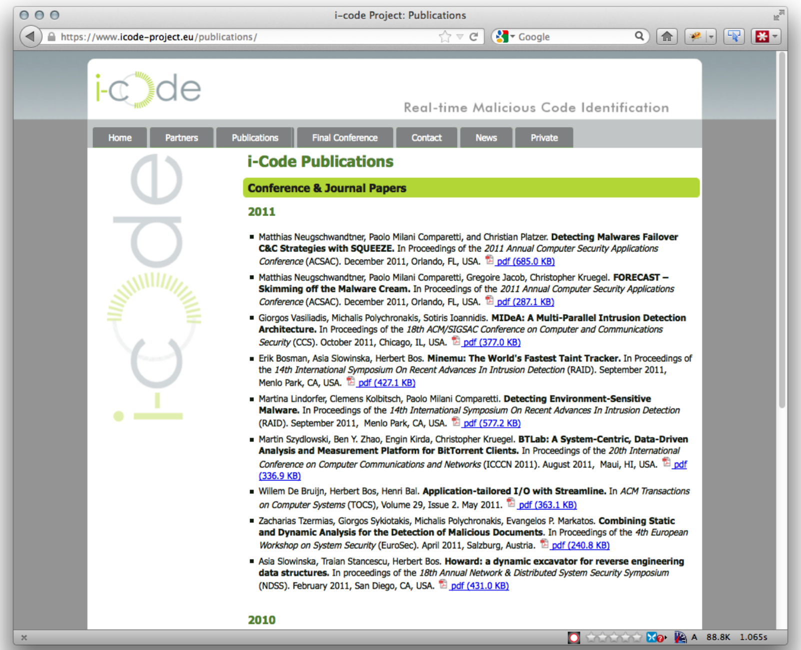
- Article in ERCIM News (July 2012) "i-Code: Real-Time Malicious Code Identification"
- Article in de NRC (Dutch): http://content.nrccarriere.nl/2012/06/wanted-soldaten-tegen...
- Interview in de Automatisering Gids (Dutch): http://www.automatiseringgids.nl/achtergrond/2012/03/we...
- Televison interview on Channel 4 News: http://www.channel4.com/news/anonymous-ethical-hackers-or...
- Article in The Economist (Greek Edition) "From malicious software to malicious documents": http://www.icode-project.eu/media/page-media/2011/7/29/...

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 the project publications, can be seen in Figure 1.

The website 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. But as we can see in Figure 2 i-Code also attracted interest from China and India.

The number of pageviews per month can be seen in Figure 3. We can see a big boost in the number of viewed pages in the last month of the project. This can be attributed to the interest sparked by the promotion of the i-Code Final Conference.



i–code Project: Publications

https://www.icode-project.eu/publications/

Real-time Malicious Code Identification

Home | Partners | Publications | Final Conference | Contact | News | Private

i-Code Publications

Conference & Journal Papers

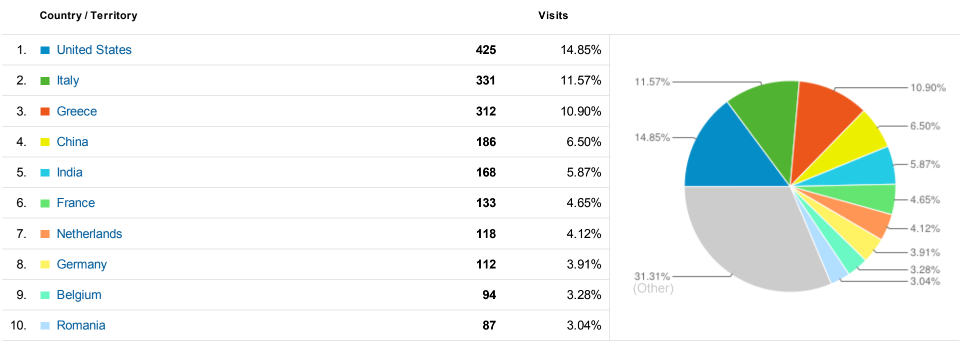
2011

- Matthias Neugschwandtner, Paolo Milani Comparetti, and Christian Platzer. **Detecting Malwares Failover C&C Strategies with SQUEEZE.** In Proceedings of the *2011 Annual Computer Security Applications Conference* (ACSAC). December 2011, Orlando, FL, USA. pdf (685.0 KB)
- Matthias Neugschwandtner, Paolo Milani Comparetti, Gregoire Jacob, Christopher Kruegel. **FORECAST – Skimming off the Malware Cream.** In Proceedings of the *2011 Annual Computer Security Applications Conference* (ACSAC). December 2011, Orlando, FL, USA. pdf (287.1 KB)
- Giorgos Vasiliadis, Michalis Polychronakis, Sotiris Ioannidis. **MIDeA: A Multi-Parallel Intrusion Detection Architecture.** In Proceedings of the *18th ACM/SIGSAC Conference on Computer and Communications Security* (CCS). October 2011, Chicago, IL, USA. pdf (377.0 KB)
- Erik Bosman, Asia Slowinska, Herbert Bos. **Minemu: The World's Fastest Taint Tracker.** In Proceedings of the *14th International Symposium On Recent Advances In Intrusion Detection* (RAID). September 2011, Menlo Park, CA, USA. pdf (427.1 KB)
- Martina Lindorfer, Clemens Kolbitsch, Paolo Milani Comparetti. **Detecting Environment-Sensitive Malware.** In Proceedings of the *14th International Symposium On Recent Advances In Intrusion Detection* (RAID). September 2011, Menlo Park, CA, USA. pdf (577.2 KB)
- Martin Szydlowski, Ben Y. Zhao, Engin Kirda, Christopher Kruegel. **BTLab: A System-Centric, Data-Driven Analysis and Measurement Platform for BitTorrent Clients.** In Proceedings of the *20th International Conference on Computer Communications and Networks* (ICCCN 2011). August 2011, Maui, HI, USA. pdf (336.9 KB)
- Willem De Bruijn, Herbert Bos, Henri Bal. **Application-tailored I/O with Streamline.** In *ACM Transactions on Computer Systems* (TOCS), Volume 29, Issue 2. May 2011. pdf (363.1 KB)
- Zacharias Tzermias, Giorgos Sykiotakis, Michalis Polychronakis, Evangelos P. Markatos. **Combining Static and Dynamic Analysis for the Detection of Malicious Documents.** In Proceedings of the *4th European Workshop on System Security* (EuroSec). April 2011, Salzburg, Austria. pdf (240.8 KB)
- Asia Slowinska, Traian Stancescu, Herbert Bos. **Howard: a dynamic excavator for reverse engineering data structures.** In proceedings of the *18th Annual Network & Distributed System Security Symposium* (NDSS). February 2011, San Diego, CA, USA. pdf (431.0 KB)

2010

A 88.8K 1.065s

Figure 1: A screenshot of a page from the project's website.



	Country / Territory	Visits	
1.	🟦 United States	425	14.85%
2.	🟩 Italy	331	11.57%
3.	🟧 Greece	312	10.90%
4.	🟨 China	186	6.50%
5.	🟦 India	168	5.87%
6.	🟩 France	133	4.65%
7.	🟧 Netherlands	118	4.12%
8.	🟨 Germany	112	3.91%
9.	🟩 Belgium	94	3.28%
10.	🟦 Romania	87	3.04%

Figure 2: Website visitors per country. There were a total of 2,862 visits in total, i.e., visitors viewed an average of 2.55 pages.

5 Final Conference

The i-Code Final Conference was held on June 27th, 2012 in Brussels; the Sheraton Brussels Airport Hotel was chosen as location because it was the most convenient location for the attendees to fly in and out in the same day.

The conference was organized as an open event, in order to let anyone who was interested to join freely after registering. At the end there was a total of almost 40 attendees (mainly from Europe), showing that the general interest on the topics we worked on in the i-Code project is generally high.

The event lasted the entire day and consisted of two sessions: the morning session was dedicated to the i-Code project and all the partners presented their work; in the afternoon session, instead, members of the European digital forensics community talked about their works and experiences. Table 1 reports the conference schedule.

The i-Code Final Conference was a great opportunity to both present the work done for the project itself and to promote networking among people involved in the same scientific subjects, to let them discuss about the delivered talks and possibly create new interesting scientific opportunities. Overall, the conference was a success and many positive feedbacks were collected from the attendees.

Figures 4, 5 and 6 show some pictures of the event.

Time	Talk and speaker
10.00 - 10.05	**Welcome and introduction** *Stefano Zanero, Politecnico di Milano*
10.05 - 10.30	**The i-Code forensic console: description and demonstration** *Alessandro Frossi, Politecnico di Milano*
10.30 - 10.50	**From Shellcode to Return–Oriented Programming: Detecting Malicious Code using Code Emulation** *Michalis Polychronakis, Columbia University*
11.10 - 11.30	**Argos and its applications** *Remco Vermeulen, Vrije Universiteit Amsterdam*
11.30 - 11.50	**AccessMiner: Using System-Centric Models for Malware Protection** *Davide Balzarotti, Institut Eurcom*
11.50 - 12.10	**Anubis: Analyzing Unknown Binaries** *Martina Lindorfer, Vienna University of Technology*
12.10 - 12.30	**Q&A on i-Code** *Stefano Zanero, Politecnico di Milano*
13.30 - 14.00	**Digital forensic in the real world** *Erwin van Eijck, NFI*
14.00 - 14.30	**HyperDbg: kernel debugging thourgh hardware–assisted virtualization** *Aristide Fattori, Universit degli Studi di Milano*
14.30 - 15.00	**Recent advances in Suricata** *Victor Julien, Suricata*
15.15 - 15.45	**HoneySpider Network 2.0: Detecting client-side threats the easy way** *Piotr Kijewski, NASK*
15.45 - 16.15	**BANOMAD** *Julio Canto, VirusTotal.com*
16.15 - 17.00	**Final discussion / roundtable**

Table 1: i-Code Final Conference schedule

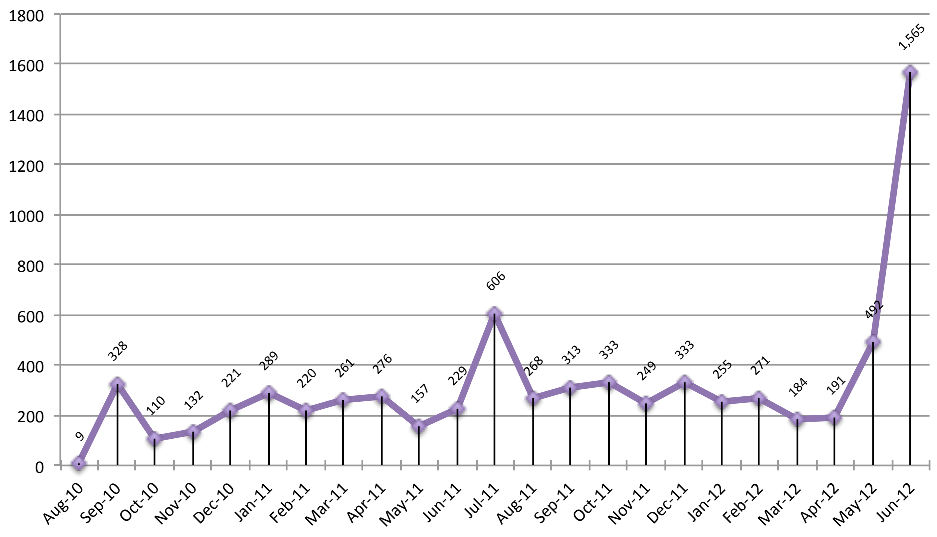


Figure 3: Pageviews per month for the projecect website. A total of 7,292 pages were viewed through the duration of the project.

i-Code Project
Final Conference
i-Code
www.icode-project.eu
MORNING SESSION
THE i-CODE PROJECT
Chair Stefano Zanero, Politecnico di Milano (workshop chair)
10.00 Welcome and introduction
Stefano Zanero, Politecnico di Milano (workshop chair)
10.05 The i-Code forensic console:description and demonstration
Alessandro Frossi, Politecnico di Milano
10.30 From Shellcode to Return-Oriented Programming: Detecting Malicious Code using Code Emulation
Michalis Polychronakis, Columbia University
10.50 Coffee Break
11.10 Argos and its applications
Remco Vermeulen, Vrije Universiteit Amsterdam
11.30 AccessMiner: Using System-Centric Models for Malware Protection
Davide Balzarotti, Institut Eurécom
11.50 Anubis: Analyzing Unknown Binaries
Martina Lindorfer, Vienna University of Technology
12.10 Q&A on i-Code
Stefano Zanero, Politecnico di Milano (workshop chair)
12.30 LUNCH
AFTERNOON SESSION
THE FORENSICS COMMUNITY IN EUROPE
Chair Herbert Bos, Vrije Universiteit Amsterdam
13.30 Digital forensics in the real world
Erwin van Eijck, NFI
14.00 HyperDbg: kernel debugging through hardware-assisted virtualization
Aristide Fattori, Università degli Studi di Milano
14.30 Recent advances in Suricata
Victor Julien, Suricata
15.00 Coffee Break
15.15 HoneySpider Network 2.0: Detecting client-side threats the easy way
Piotr Kijewski, NASK
15.45 Banomad
Julio Canto, VirusTotal.com
16.15 Final discussion/roundtable

Figure 4: i-Code Final Conference location entrance

Figure 5: Introductory talk for i-Code Final Conference

Figure 6: Promoting networking during i-Code Final Conference