EDITORES:

Manuel A. Serrano - Eduardo Fernández-Medina

Cristina Alcaraz - Noemí de Castro - Guillermo Calvo

Actas de las VI Jornadas Nacionales
(JNIC2021 LIVE)

INVESTIGACIÓN EN ciber segu ridad

Ediciones de la Universidad
de Castilla-La Mancha

Investigación en Ciberseguridad

Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)

Online 9-10 de junio de 2021
Universidad de Castilla-La Mancha

Investigación en Ciberseguridad

Actas de las VI Jornadas Nacionales (JNIC2021 LIVE)

Online 9-10 de junio de 2021
Universidad de Castilla-La Mancha

Editores:
Manuel A. Serrano,
Eduardo Fernández-Medina,
Cristina Alcaraz
Noemí de Castro
Guillermo Calvo

Ediciones de la Universidad
de Castilla-La Mancha

Cuenca, 2021




Edita: Ediciones de la Universidad de Castilla-La Mancha

Colección JORNADAS Y CONGRESOS n.º 34

Esta editorial es miembro de la UNE, lo que garantiza la difusión y comercialización de sus publicaciones a nivel nacional e internacional.

I.S.B.N.: 978-84-9044-463-4
D.O.I.: http://doi.org/10.18239/jornadas_2021.34.00

Hecho en España (U.E.) – *Made in Spain (E.U.)*

Bienvenida del Comité Organizador

Tras la parada provocada por la pandemia en 2020, las VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) vuelven el 9 y 10 de Junio del 2021 con energías renovadas, y por primera vez en su historia, en un formato 100% online. Esta edición de las JNIC es organizada por los grupos GSyA y Alarcos de la Universidad de Castilla-La Mancha en Ciudad Real, y con la activa colaboración del comité ejecutivo, de los presidentes de los distintos comités de programa y del Instituto Nacional de Ciberseguridad (INCIBE). Continúa de este modo la senda de consolidación de unas jornadas que se celebraron por primera vez en León en 2015 y le siguieron Granada, Madrid, San Sebastián y Cáceres, consecutivamente hasta 2019, y que, en condiciones normales se habrían celebrado en Ciudad Real en 2020.

Estas jornadas se han convertido en un foro de encuentro de los actores más relevantes en el ámbito de la ciberseguridad en España. En ellas, no sólo se presentan algunos de los trabajos científicos punteros en las diversas áreas de ciberseguridad, sino que se presta especial atención a la formación e innovación educativa en materia de ciberseguridad, y también a la conexión con la industria, a través de propuestas de transferencia de tecnología. Tanto es así que, este año se presentan en el Programa de Transferencia algunas modificaciones sobre su funcionamiento y desarrollo que han sido diseñadas con la intención de mejorarlo y hacerlo más valioso para toda la comunidad investigadora en ciberseguridad.

Además de lo anterior, en las JNIC estarán presentes excepcionales ponentes (Soledad Antelada, del Lawrence Berkeley National Laboratory, Ramsés Gallego, de Micro Focus y Mónica Mateos, del Mando Conjunto de Ciberdefensa) mediante tres charlas invitadas y se desarrollarán dos mesas redondas. Éstas contarán con la participación de las organizaciones más relevantes en el panorama industrial, social y de emprendimiento en relación con la ciberseguridad, analizando y debatiendo el papel que está tomando la ciberseguridad en distintos ámbitos relevantes.

En esta edición de JNIC se han establecido tres modalidades de contribuciones de investigación, los clásicos artículos largos de investigación original, los artículos cortos con investigación en un estado más preliminar, y resúmenes extendidos de publicaciones muy relevantes y de alto impacto en materia de ciberseguridad publicados entre los años 2019 y 2021. En el caso de contribuciones de formación e innovación educativa, y también de transferencias se han considerado solamente artículos largos. Se han recibido para su valoración un total de 86

contribuciones organizadas en 26, 27 y 33 artículos largos, cortos y resúmenes ya publicados, de los que los respectivos comités de programa han aceptado 21, 19 y 27, respectivamente. En total se ha contado con una ratio de aceptación del 77%. Estas cifras indican una participación en las jornadas que continúa creciendo, y una madurez del sector español de la ciberseguridad que ya cuenta con un volumen importante de publicaciones de alto impacto.

El formato online de esta edición de las jornadas nos ha motivado a organizar las jornadas de modo más compacto, distinguiendo por primera vez entre actividades plenarias (charlas invitadas, mesas redondas, sesión de formación e innovación educativa, sesión de transferencia de tecnología, junto a inauguración y clausura) y sesiones paralelas de presentación de artículos científicos. En concreto, se han organizado 10 sesiones de presentación de artículos científicos en dos líneas paralelas, sobre las siguientes temáticas: detección de intrusos y gestión de anomalías (I y II), ciberataques e inteligencia de amenazas, análisis forense y cibercrimen, ciberseguridad industrial, inteligencia artificial y ciberseguridad, gobierno y riesgo, tecnologías emergentes y entrenamiento, criptografía, y finalmente privacidad.

En esta edición de las jornadas se han organizado dos números especiales de revistas con elevado factor de impacto para que los artículos científicos mejor valorados por el comité de programa científico puedan enviar versiones extendidas de dichos artículos. Adicionalmente, se han otorgado premios al mejor artículo en cada una de las categorías. En el marco de las JNIC también hemos contado con la participación de la Red de Excelencia Nacional de Investigación en Ciberseguridad (RENIC), impulsando la ciberseguridad a través de la entrega de los premios al *Mejor Trabajo Fin de Máster en Ciberseguridad* y a la *Mejor Tesis Doctoral en Ciberseguridad*. También se ha querido acercar a los jóvenes talentos en ciberseguridad a las JNIC, a través de un CTF (Capture The Flag) organizado por la Universidad de Extremadura y patrocinado por Viewnext.

Desde el equipo que hemos organizado las JNIC2021 queremos agradecer a todas aquellas personas y entidades que han hecho posible su celebración, comenzando por los autores de los distintos trabajos enviados y los asistentes a las jornadas, los tres ponentes invitados, las personas y organizaciones que han participado en las dos mesas redondas, los integrantes de los distintos comités de programa por sus interesantes comentarios en los procesos de revisión y por su colaboración durante las fases de discusión y debate interno, los presidentes de las sesiones, la Universidad de Extremadura por organizar el CTF y la empresa Viewnext por patrocinarlo, los técnicos del área TIC de la UCLM por el apoyo con la plataforma de comunicación, los voluntarios de la UCLM y al resto de organizaciones y entidades patrocinadoras, entre las que se encuentra la Escuela Superior de Informática, el Departamento de Tecnologías y Sistemas de Información y el Instituto de Tecnologías y Sistemas de Información, todos ellos de la Universidad de Castilla-La Mancha, la red RENIC, las cátedras (Telefónica e Indra) y aulas (Avanttic y Alpinia) de la Escuela Superior de Informática, la empresa Cojali, y muy especialmente por su apoyo y contribución al propio INCIBE.

Manuel A. Serrano, Eduardo Fernández-Medina
Presidentes del Comité Organizador
Cristina Alcaraz
Presidenta del Comité de Programa Científico
Noemí de Castro
Presidenta del Comité de Programa de Formación e Innovación Educativa
Guillermo Calvo Flores
Presidente del Comité de Transferencia Tecnológica

Índice General

Comité Ejecutivo

Juan Díez González	INCIBE
Luis Javier García Villalba	Universidad de Complutense de Madrid
Eduardo Fernández-Medina Patón	Universidad de Castilla-La Mancha
Guillermo Suárez-Tangil	IMDEA Networks Institute
Andrés Caro Lindo	Universidad de Extremadura
Pedro García Teodoro	Universidad de Granada. Representante de red RENIC
Noemí de Castro García	Universidad de León
Rafael María Estepa Alonso	Universidad de Sevilla
Pedro Peris López	Universidad Carlos III de Madrid

Comité Organizador

Presidentes del Comité Organizador

Eduardo Fernández-Medina Patón	Universidad de Castilla-la Mancha
Manuel Ángel Serrano Martín	Universidad de Castilla-la Mancha

Finanzas

David García Rosado	Universidad de Castilla-la Mancha
Luis Enrique Sánchez Crespo	Universidad de Castilla-la Mancha

Actas

Antonio Santos-Olmo Parra	Universidad de Castilla-la Mancha

Difusión

Julio Moreno García-Nieto	Universidad de Castilla-la Mancha
José Antonio Cruz Lemus	Universidad de Castilla-la Mancha
María A Moraga de la Rubia	Universidad de Castilla-la Mancha

Webmaster

Aurelio José Horneros Cano	Universidad de Castilla-la Mancha

Logística y Organización

Ignacio García-Rodriguez de Guzmán	Universidad de Castilla-la Mancha
Ismael Caballero Muñoz-Reja	Universidad de Castilla-la Mancha
Gregoria Romero Grande	Universidad de Castilla-la Mancha
Natalia Sanchez Pinilla	Universidad de Castilla-la Mancha

Comité de Programa Científico

Presidenta

Cristina Alcaraz Tello	Universidad de Málaga

Miembros

Aitana Alonso Nogueira	INCIBE
Marcos Arjona Fernández	ElevenPaths
Ana Ayerbe Fernández-Cuesta	Tecnalia
Marta Beltrán Pardo	Universidad Rey Juan Carlos
Carlos Blanco Bueno	Universidad de Cantabria
Jorge Blasco Alís	Royal Holloway, University of London
Pino Caballero-Gil	Universidad de La Laguna
Andrés Caro Lindo	Universidad de Extremadura
Jordi Castellà Roca	Universitat Rovira i Virgili
José M. de Fuentes García-Romero de Tejada	Universidad Carlos III de Madrid
Jesús Esteban Díaz Verdejo	Universidad de Granada
Josep Lluis Ferrer Gomila	Universitat de les Illes Balears
Dario Fiore	IMDEA Software Institute
David García Rosado	Universidad de Castilla-La Mancha
Pedro García Teodoro	Universidad de Granada
Luis Javier García Villalba	Universidad Complutense de Madrid
Iñaki Garitano Garitano	Mondragon Unibertsitatea
Félix Gómez Mármol	Universidad de Murcia
Lorena González Manzano	Universidad Carlos III de Madrid
María Isabel González Vasco	Universidad Rey Juan Carlos I
Julio César Hernández Castro	University of Kent
Luis Hernández Encinas	CSIC
Jorge López Hernández-Ardieta	Banco Santander
Javier López Muñoz	Universidad de Málaga
Rafael Martínez Gasca	Universidad de Sevilla
Gregorio Martínez Pérez	Universidad de Murcia

David Megías Jiménez	Universitat Oberta de Cataluña
Luis Panizo Alonso	Universidad de León
Fernando Pérez González	Universidad de Vigo
Aljosa Pasic	ATOS
Ricardo J. Rodríguez	Universidad de Zaragoza
Fernando Román Muñoz	Universidad Complutense de Madrid
Luis Enrique Sánchez Crespo	Universidad de Castilla-La Mancha
José Soler	Technical University of Denmark-DTU
Miguel Soriano Ibáñez	Universidad Politécnica de Cataluña
Victor A. Villagrá González	Universidad Politécnica de Madrid
Urko Zurutuza Ortega	Mondragon Unibertsitatea
Lilian Adkinson Orellana	Gradiant
Juan Hernández Serrano	Universitat Politécnica de Cataluña

Comité de Programa de Formación e Innovación Educativa

Presidenta

Noemí De Castro García	Universidad de León

Miembros

Adriana Suárez Corona	Universidad de León
Raquel Poy Castro	Universidad de León
José Carlos Sancho Núñez	Universidad de Extremadura
Isaac Agudo Ruiz	Universidad de Málaga
Ana Isabel González-Tablas Ferreres	Universidad Carlos III de Madrid
Xavier Larriva	Universidad Politécnica de Madrid
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Lorena González Manzano	Universidad Carlos III de Madrid
María Isabel González Vasco	Universidad Rey Juan Carlos
David García Rosado	Universidad de Castilla - La Mancha
Sara García Bécares	INCIBE

Comité de Transferencia Tecnológica

Presidente

Guillermo Calvo Flores	INCIBE

Miembros

José Luis González Sánchez	COMPUTAEX
Marcos Arjona Fernández	ElevenPaths
Victor Villagrá González	Universidad Politécnica de Madrid
Luis Enrique Sánchez Crespo	Universidad de Castilla – La Mancha

http://doi.org/10.18239/jornadas_2021.34.26

A Review of "Pre-processing Memory Dumps to Improve Similarity Score of Windows Modules"

Miguel Martín-Pérez
Universidad de Zaragoza, Spain
miguelmartinperez@unizar.es

Ricardo J. Rodríguez
Universidad de Zaragoza, Spain
rjrodriguez@unizar.es

Davide Balzarotti
EURECOM, France
davide.balzarotti@eurecom.fr

***Abstract*—Memory forensics is useful to provide a fast triage on running processes at the time of memory acquisition in order to avoid unnecessary forensic analysis. However, due to the effects of the execution of the process itself, traditional cryptographic hashes are unsuitable in memory forensics. Similarity digest algorithms allow an analyst to compute a similarity score of inputs that can be slightly different. In this paper, we focus on the issues caused by relocation of Windows processes and system libraries when computing similarities between them. To overcome these issues, we introduce two methods (**GUIDED DE-RELOCATION **and** LINEAR SWEEP DE-RELOCATION**) to pre-process a memory dump. The goal of both methods is to identify and undo the effect of relocation in every module contained in the dump, providing *sanitized* inputs to similarity digest algorithms that improve similarity scores between modules.** GUIDED DE-RELOCATION **relies on specific structures of the Windows PE format, while** LINEAR SWEEP DE-RELOCATION **relies on a disassembling process to identify assembly instructions having memory operands that address the memory range of the module. We have evaluated them in different scenarios. Our results demonstrate that pre-processing memory dumps with these methods significantly improves similarity scores between memory modules. In addition, both methods have been integrated in a Volatility plugin.**

***Index Terms*—similarity digest algorithms, memory forensics, Windows, relocation**

Tipo de contribución: *Investigación ya publicada en* "Pre-processing memory dumps to improve similarity score of Windows modules," Computers & Security, vol. 101, p. 102119, 2021 [1].

I. EXTENDED ABSTRACT

Memory forensics is a branch of the computer forensics process, normally carried out as part of the detection and analysis stage in an incident response process [2]. In particular, memory forensics, unlike disk forensics, deals with the recovery of digital evidence from computer memory instead of computer storage media. Furthermore, the initial triage in memory forensics is faster than in persistent storage forensics since the quantity of data to be analyzed is smaller.

A memory forensic analyst can triage the list of processes running at the acquisition time to discard well-known processes or to focus her attention on particular ones. Thus, she needs some way to identify processes. In disk forensics, cryptographic hash (one-way) functions [3] such as MD5, SHA-1, or SHA-256 functions are commonly used for data integrity and file identification of a seized device [4]. A desirable property of any cryptographic hash function is the avalanche effect property [5], which guarantees that the hash values of two similar, but not identical, inputs produce radically different outputs. Due to this property, these crypto hash functions are unsuitable for identifying common processes that belong to the same binary application, but in different executions.

A common pitfall is to think that the content of a running process and its corresponding executable file are identical. In fact, the OS loader may apply a number of transformations when the executable file is mapped into memory. For instance, software defense techniques such as Address Space Layout Randomization ensure that executable files are mapped in memory regions that are different among consecutive executions or among consecutive system reboots (in Windows systems). Furthermore, the size of the executable file in the memory may be larger than on disk due to memory alignment issues, as the granularity of the memory subsystem OS manager determines the minimum quantity of allocated memory space (for instance, 4 KiB in Windows, macOS, and GNU/Linux).

To overcome these limitations, approximate matching or *similarity digest algorithms* (SDA) have emerged in recent years as a prominent approach that is more robust against active adversaries than traditional hashing [4]. SDA identify similarities between two digital artifacts providing a measure of similarity, normally in the range of $[0, 100]$. This similarity score enables an analyst to find out whether artifacts resemble each other or whether an artifact is contained in another artifact [6].

As mentioned above, the differences between processes are mainly motivated by the work of the relocation process. These differences, in turn, negatively affect the similarity scores provided by the similarity digest algorithms (in some cases even resulting in a similarity close to zero).

To minimize the effect of these differences, in this paper we propose two methods to process the input given to a similarity digest algorithm prior to computing its similarity hash. Both pre-processing methods undo the work performed by the relocation process, but in different ways: the method called GUIDED DE-RELOCATION relies on particular kernel-space structures that might be contained in the memory dump. These structures point to the affected bytes and allow a precise de-relocation.

While the LINEAR SWEEP DE-RELOCATION method performs a linear sweep disassembly of the binary code of a process. Specifically, it identifies all possible sub-structures of the PE format and then it performs a linear sweep disassembly of the unidentified bytes, selecting the longest

sequence of instructions as the most probable correct sequence. As the last step, instructions with operands that point to module space are normalized.

Both methods work with small memory page granularity (4 KiB). We have evaluated them by comparing the similarity scores generated by the `dcfldd`, `ssdeep`, `sdhash`, and `TLSH` SDAs, and shown that the similarity score is improved when any method is used. Figure 1 shows the similarity scores between related pages when no de-relocation is performed for 32-bit architecture, which has the worst similarity result. Similarly, Figure 2 and 3 show the improvement of the similarity when the GUIDED DE-RELOCATION method and the LINEAR SWEEP DE-RELOCATION method are applied, respectively. The results in the latter case are worse than the former one because LINEAR SWEEP DE-RELOCATION identifies many, but not all, of the bytes affected by relocation. Nevertheless, the improvement of the results shown is enough to identify successfully similar modules. Last, in Figure 4 we display the result of comparing modules processed with different methods, showing that both methods are compatible and the results are equal to the worst case. On the contrary, the results of applying de-relocation methods in 64-bit architectures are very similar to the results without any pre-processing. This is motivated because in 64-bit mode the RIP-relative addressing form was introduced, which facilitates the construction of position-independent code and therefore the bytes affected by relocation will be only the ones related to function addresses of shared libraries. We refer the reader to [1] for more details in this issue.

In addition, we have also evaluated to what extent the similarity score of each algorithm considered in the paper is affected by the loading process. We found that these algorithms are particularly sensitive to byte modifications, and that *intelligent* byte modifications can dramatically affect the similarity score for some of these algorithms.

We have developed a Volatility plugin that implements the de-relocation methods and the SDA considered in this paper. For the sake of open science, we have released it under the GNU/GPLv3 license [7]. The tool is designed in an extensible way, allowing for quick additions of SDAs.

The full version of this paper (with a full description of the experiments and limitations) was published in [1].

Figure 1. Not pre-processed 32-bit architecture

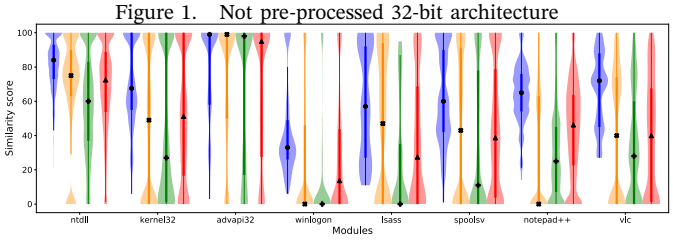


Figure 2. GUIDED DE-RELOCATION 32-bit architecture

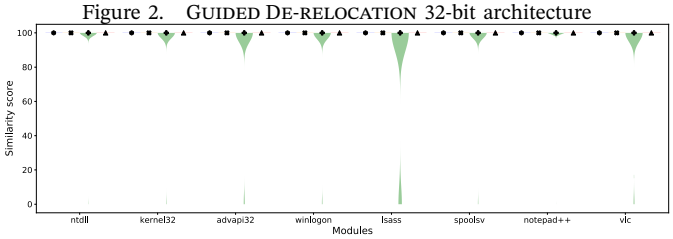


Figure 3. LINEAR SWEEP DE-RELOCATION 32-bit architecture

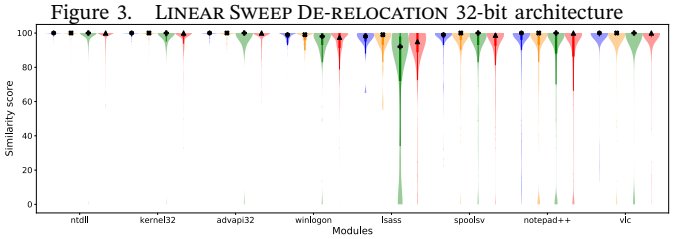
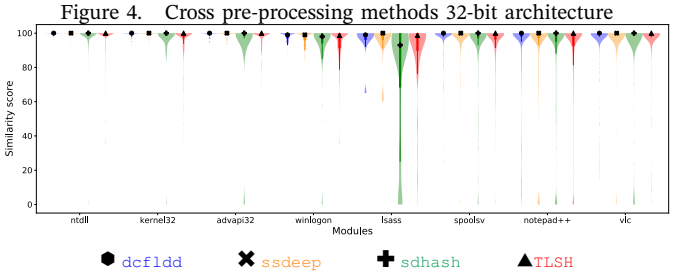


Figure 4. Cross pre-processing methods 32-bit architecture



ACKNOWLEDGEMENTS

The research was supported by the Spanish Ministry of Science, Innovation and Universities under grant MEDRESE-RTI2018-098543-B-I00 and by the University, Industry and Innovation Department of the Aragonese Government under *Programa de Proyectos Estratégicos de Grupos de Investigación* (DisCo research group, ref. T21-20R). It was also supported by the Spanish National Cybersecurity Institute (INCIBE) "Ayudas para la excelencia de los equipos de investigación avanzada en ciberseguridad", grant numbers INCIBEC-2015-02486 and INCIBEI-2015-27300. This work was also supported in part by the European Research Council (ERC) under the European Union Horizon 2020 research and innovation programme (grant agreement No 771844 BitCrumbs). This research has been developed during a short-research term in EURECOM supported by *Campus de Excelencia Internacional del Valle del Ebro* (Campus Iberus), "Convenio de subvención Erasmus+ Educación Superior para prácticas Consorcio Iberus+", and *Universidad de Zaragoza, Fundación Bancaria Ibercaja y Fundación CAI* "Programa Ibercaja-CAI de Estancias de Investigación", grant number IT 7/19.

REFERENCES

[1] M. Martín-Pérez, R. J. Rodríguez, and D. Balzarotti, "Pre-processing memory dumps to improve similarity score of Windows modules," *Computers & Security*, vol. 101, p. 102119, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820303928

[2] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer Security Incident Handling Guide," National Institute of Standards and Technology (NIST), techreport SP 800-61 Rev. 2, Sep. 2012, special Publication (NIST SP).

[3] O. Goldreich, *Foundations of Cryptography: Volume 1*. New York, NY, USA: Cambridge University Press, 2006.

[4] V. S. Harichandran, F. Breitinger, and I. Baggili, "Bytewise Approximate Matching: the Good, the Bad, and the Unknown," *Journal of Digital Forensics, Security and Law*, vol. 11, no. 2, 2016.

[5] A. F. Webster and S. E. Tavares, "On the Design of S-Boxes," in *Advances in Cryptology — CRYPTO '85 Proceedings*, H. C. Williams, Ed. Springer Berlin Heidelberg, 1986, pp. 523–534.

[6] F. Breitinger, B. Guttman, M. McCarrin, V. Roussev, and D. White, "Approximate Matching: Definition and Terminology," National Institute of Standards and Technology, techreport NIST Special Publication 800-168, May 2014.

[7] M. Martín-Pérez, "Similarity Unrelocated Module Volatility plugin," [Online; https://github.com/reverseame/similarity-unrelocated-module], Jul. 2020.