RAMSES

Internet Forensic platform for tracking the money flow of financially-motivated malware

H2020 - 700326

D6.1 Design of the analysis system and specifications

Lead Author: POLIMI
With contributions from: UNIKENT
Reviewer: USAAR

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	August 2017
Actual delivery date:	31/08/2017
Version:	1.0
Total number of pages:	33
Keywords:	Financial Trojas, Bitcoin, Forensic, Intelligence

Abstract

This document describes the design and specification of the analysis systems that the RAMSES platform will use to analyze financially-motivated malware sample and the money flows related to their malicious activity.

This system is composed by two different tools: a memory forensics tool for banking trojan analysis and detection, and a framework for extracting intelligence from the Bitcoin network.

Executive summary

The goal of this document is to describe the design and specification of the analysis systems required for the RAMSES platform. The analysis systems have been designed taking into account the needs of the Law Enforcement Agencies (LEAs) as well as the feedback provided by other members of the consortium, described in the deliverables *D.2.2 Report on relevant scenarios* [1] and *D.2.3 Report on user requirement for the design of LEAs tools* [2].

We designed two main systems:

Prometheus: A memory forensics tool for banking trojan analysis and detection. Prometheus is an automatic system that is able to analyze banking trojans that base their attack technique on DOM (Document Object Model) modifications. Trojans leave artifacts of the injection behaviour in the infected machine's memory, e.g., list of targets URLs. Prometheus, leveraging memory forensics techniques, is able to inspect memory and extract these artifacts that can be used as indicators of compromise.

BitIodine: A tool for extracting intelligence from the Bitcoin network. BitIodine is a modular framework which parses the blockchain, clusters address that are likely to belong to a same user or group of users, classifies such users and labels them, and finally visualizes complex information extracted from the Bitcoin network.

In order to describe specification of such systems, this document provides a description of the requirements, the design choices, and the technology used for the development.

Document Information

IST Project Number	700326	Acronym	RAMSES
Full Title	Internet Forensic platform for tracking the money flow of financially-motivated malware		
Project URL	http://www.ramses2020.eu		
EU Project Officer	Nada Milisavljevic		

Deliverable	Number	D6.1	Title	Design of the analysis system and specifications
Work Package	Number	WP6	Title	Forensic analysis of malware monetization techniques

Date of Delivery	Contractual	M12	Actual	M12
Status	version 0.1		final □	
Nature	prototype □ report X demonstrator □ other □			
Dissemination level	public X restricted □			

Authors (Partner)	POLIMI			
Responsible Author	Name	Stefano Zanero	E-mail	stefano.zanero@polimi.it
	Partner	POLIMI	Phone	

Abstract (for dissemination)	This document describes the design and specification of the analysis systems that the RAMSES platform will use to analyze financially-motivated malware sample and the money flows related to their malicious activity. This system is composed by two different tools: a memory forensics tool for banking trojan analysis and detection, and a framework for extracting intelligence from the Bitcoin network.
Keywords	

Version Log			
Issue Date	Rev. No.	Author	Change
31/05/2017	0.1	Stefano Zanero	Initial proposed TOC
15/07/2017	0.2	Michele Carminati	Revised TOC
10/08/2017	0.2	Michele Carminati	First Draft
15/08/2017	0.3	Andrea Continella	Review and feedback
23/08/2017	0.4	Andrea Continella	Second Draft
25/08/2017	0.5	Michele Carminati	Review and feedback
29/08/2017	0.5	Michael Brengel	Internal review
29/08/2017	0.6	Andrea Continella	Revision on the basis of Internal review
29/08/2017	0.7	Michele Carminati	Review and Integration of UoK contribution
30/08/2017	1.0	POLIMI Team	Final Version

Table of Contents

List of figures and/or list of tables

Abbreviations

API: Application Programming Interface

CIA: Confidentiality, Integrity and Availability

C&C servers: Command and Control servers

DDoS: Distributed Denegation of Service

DOM: Document Object Model

HTTPS: HyperText Transfer Protocol over Secure Socket Layer

ICT: Information and Communication Technologies

IOC: Indicator of Compromise

IP: Internet Protocol

LEAs: Law Enforcement Agencies

MitB: Man-in-the-Browser

OSINT: Open Source Intelligence

PRNU: Photo Response Non-Uniformity

SaaS: Software as a Service

SSL: Secure Sockets Layer

SVM: Support Vector Machine

URL: Uniform Resource Locator

Definitions

Bitcoin: The most prevalent cryptocurrency currently in existence. Bitcoin represents the first successful cryptocurrency and has mass media exposure. It is the most common currency for ransomware to request as payment, both due to its popularity (making it likely that a victim will have heard of it) and the wealth of information and sources that make purchasing Bitcoin less arduous than other cryptocurrencies.

Blockchain: A fundamental technology that enables cryptocurrencies. A blockchain is effectively a distributed ledger, which provides a record of all transactions that have been agreed on by a consensus among trusted nodes on a network. The most common example is Satoshi's Bitcoin blockchain, but other examples are common.

Crypto-Ransomware: A specific form of ransomware, which works by encrypting the contents of the target computer and using the decryption key as a bargaining chip. Differs from some strains of ransomware by focusing on threats of implicit data-loss instead of other forms of control. It may still rely on associated strategies, such as file deletion, but passively threatens to leave files encrypted in a manner that would make them irretrievable without the appropriate key. Generally a cash fee is asked (though some have more exotic requirements such as pyramid schemes).

Cryptocurrency: Digital currency, backed by one of a variety of means and using blockchain technology to provide a means of undisputed exchange of currency. Usually has a focus on pseudonymity community and/or niche markets that provide backing (futures, computational power etc.).

Dark-Net: A network overlaying the internet. It can only be accessed with specific software and operates using non-standard communication protocols. Usually intended to be private and anonymous, these networks are attractive to criminals and play host to black markets such as Alphabay.

Deep Web: the Deep Web (also called Invisible Web, or Hidden Web) are parts of the World Wide Web whose contents are not indexed by standard search engines for any reason. It is estimated that the Deep Web makes up 96% of the whole internet.

Malware: Malicious software. Programs that cause damage and/or disruption to a target. May involve deletion of files, spying software, ransomware or one of many other forms of attack. Some extreme examples may focus on destruction of hardware (Stuxnet).

Malware as a Service: The sale of malware and expertise, instead of direct use. Involves technically adept groups and individuals exchanging their outputs for money.

Man-in-the-Browser: A form of Internet threat related to Man-in-the-Middle (MitM). Infects a web browser by taking advantage of vulnerabilities in browser security to modify web pages, modify transaction content or insert additional transactions, all in a completely covert fashion invisible to both the user and host web application.

Ransomware: Software that focuses on seizing control of technology, software or data. The control is used as leverage in a demand for the target to exchange a sum of money for the promised (but debatable) return of the seized items. Used colloquially to reference **Crypto-ransomware**, ransomware can mean any software that focuses on seizure of assets, not just cryptographic methods.

Ransomware as a Service: The act of developing, maintaining and providing technical expertise in the use of ransomware for profit. Instead of using their software themselves, individuals and groups performing ransomware as a service sell their capabilities and outputs to others, who will then go on to use them. This may technically include distribution methods, such as botnets, which are used to send malicious software to potential victims, though this may also fall under malware as a service (depending on the goal of the purchasing party).

1 Introduction

Within the context of WP6, the RAMSES consortium aims to design, implement and make available for the research and forensic community a collection of systems that can automatically analyze malware and their monetization techniques.

The systems will specifically address two foremost use-cases of financially-motivated malware (banking trojans and ransomware) and address the malicious abuse of cryptocurrencies (Bitcoin, and any other cryptocurrencies that foreseeably will be used by malicious actors). The need for these tools and the background about such abuses will be reviewed in Section 2, where we will also review the current state-of-the-art tools available to investigators for dealing with these issues.

This document details the global design of our system(s), and the process we followed to design them.

In order to properly serve our target end-users (in particular the Law Enforcement and forensic sector) we ran a requirements elicitation process, described in Section 3, with interviews (both in form of survey and of actual discussions), which drove our requirements analysis process. In parallel we explored the usage of other crypto-currencies besides Bitcoin, to evaluate their potential as a tool for cybercriminals, and any anecdotal evidence of their usage and abuse. To identify the relevant cryptocurrencies, we conducted a study to assess their current and future relevance in cybercrime monetization. We particularly targeted existing and new cryptocurrencies that, though less popular and posterior to bitcoin, may offer benefits to criminals in terms of anonymity or untraceability, such as anoncoin (https://anoncoin.net/), zerocoin (http://zerocoin.org/) and others. For each relevant cryptocurrency we will analyze all of the methods developed in the previous literature and in Task 6.1 for Bitcoin, and determine if and how they are applicable or extensible to work on other cryptocurrencies.

In Section 4 we first recollect all of the requirements derived in Section 3, and then propose the system design. We complete the system's description in Section 5 by describing the basic technologies used, the implementation choices followed, the parameters that each tool will require, and the results that will be produced.

The proposed solution is composed by two different tools, which make different and automated types of forensic analysis. The first tool is focused on the automatic extraction of intelligence from the bitcoin network in order to classify and visualize this information, which wouldn't be possible to carry out manually. At the same time, it is an automated analysis and intelligence tool, but also offers crowdsourcing-enabled methods for sharing human-generated intelligence and annotations. It comprises all of the blocks and tools that provide state-of-the-art deanonymization and transaction graph analysis techniques for Bitcoin and for the other cryptocurrencies determined to be relevant. It offers features (both novel and reimplemented from the state of the art) to aggregate keys into user clusters, to annotate them with information crawled from open source intelligence (OSINT) sources, to cluster transactions making bitcoin flows more evident and simpler to trace and analyse. The system is designed to be easy-to-use and reserved to the research and forensics communities (including LEAs). The second tool is able to analyze banking trojans that base their attack technique on DOM modifications. In particular, it focuses on the analysis of the memory of infected machines in order to identify and to detect banking trojans.

Finally, in Section 6 we provide conclusions and key aspects of the proposed system.

2 Motivations

In the following sections, we provide an overview of the current threat landscape that involves financially motivated malware. Specifically, we first focus on banking trojans and ransomware, describing the main peculiarities of these kinds of malware. Then, we focus on the use of cryptocurrencies in the cybercrime scenario.

2.1 Financially motivated malware: Banking Trojan and Ransomware

Financially motivated malware are growing at exponential rates, with Trojans being one of the most common and dangerous types of malware, while ransomware is becoming more targeted and sophisticated. Both banking/financial Trojans and ransomware allow attackers to monetise each infection almost directly and this is the predominant reason for their continuous spread. Their explosive growth is fuelled by the fact that basically anyone, independently from their skill level, can use them, since an active underground economy (sometimes referred to, tongue-in-cheek, as "Crime-as-a-service") provides all the required resources. For example, Goncharov [1] estimated that just the Russian underground economy is a 2.3 billion dollars' market. Lindorfer et al. [2] measured that Trojans are actively developed and maintained. These and other modern malware families live in a complex environment with development kits, web-based administration panels, builders, automated distribution networks, and easy-to-use customization procedures. The most alarming consequence is that virtually anyone can buy a malware builder from underground marketplaces and create a customized sample. Grier et al. [3] investigated the emergence of the exploit-as-a-service model, showing how attackers pay for exploit kits to infect victims and propagate their own malware through drive-by downloads. Therefore, even with little or no expertise or ability to write a malware, anyone can simply purchase these "kits" and follow detailed guides and video tutorials sold online. The Trojans samples and services available on the underground markets vary, and their price depends on the features (for instance, a new, complete version of a modern banking Trojan can cost about 3,000 US$ [4]).

Hence, financially motivated malware is a category of the cybercrime overall phenomenon which is intrinsically dynamic and it has the potential to disrupt the security of both public and private organisations, as well as the functionality and integrity of their IT infrastructures. In some cases (such as the healthcare sector, which is suffering from the increasing number of attacks), also people safety may be at risk.

The two main types of financially-motivated malware deployed nowadays are banking Trojans and ransomware. Since their characteristics radically differ, we will dedicate a separate section to the analysis of each type.

2.1.1 Banking Trojans (or Information Stealers)

A particular type of Trojans, known as Information Stealers or banking Trojans, allow malware operators to intercept sensitive data such as credentials (e.g., usernames, passwords) and credit card information.

Information stealing Trojans are a growing, sophisticated threat. The most famous example is ZeuS, from which other descendants were created. This malware is actually a binary generator, which eases the creation of customized variants. For instance, as of February 19, 2017, according to ZeuS Tracker [5], there are 8,151 distinct variants that have yet to be included in the Malware Hash Registry database [6]. This number is very typical and it is also an underestimate, limited to the binaries that are currently tracked. This high number of variants results in a low detection rate overall (40% as of the same date).

Financial Trojans quite often use man-in-the-browser (MitB) techniques to perform attacks. These techniques exploit API (Application programming interface) hooking and, as the name suggests, allow malware to be logically executed inside the web browser and to intercept all data flowing through it. Also, modern banking Trojan families commonly include a module called WebInject [7], which facilitates the

manipulation and modification of data transmitted between a web server and the browser. Once the victim is infected, the WebInject module places itself between the browser's rendering engine and the API networking functions used for sending and receiving data. By hooking high-level API communication functions in user-mode code, the Trojans can intercept data more conveniently than traditional keyloggers, as they can intercept data after being decrypted. Therefore, the WebInject module is effective even in case an HTTPS (HyperText Transfer Protocol over Secure Socket Layer) connection is used.

In the following figure (Figure 1), we show an example of a real injection.

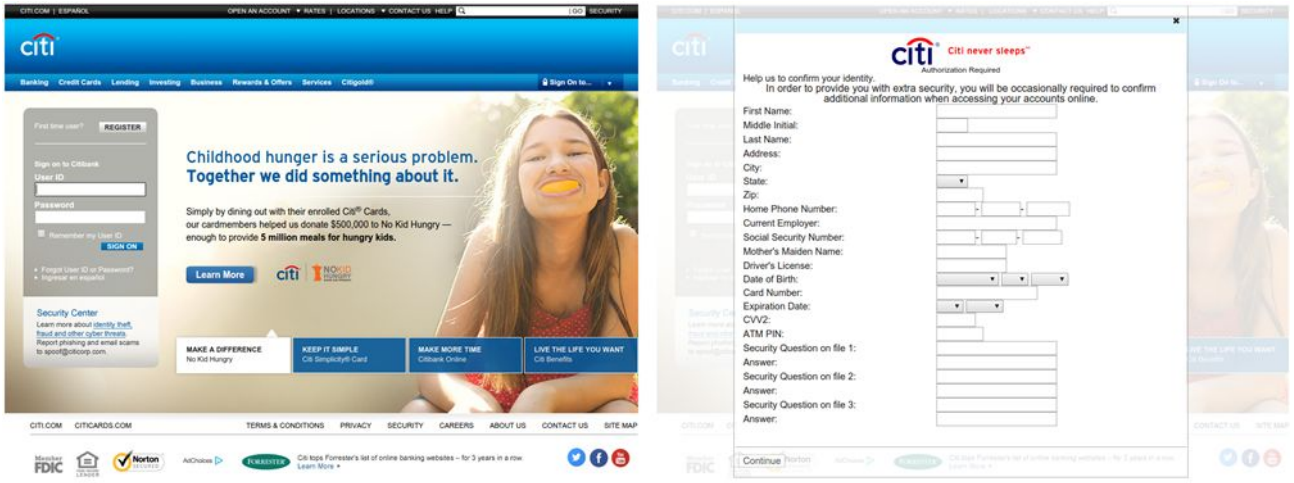

Figure 1 Example of a real injection. The screenshot on the left depicts the normal state of a banking website and the screenshot on the right shows the same banking website where a banking trojan has injected fake survey trying to steal sensitive information.

Cybercriminals can effectively inject HTML (HyperText Markup Language) code that adds extra fields in forms so as to steal sensitive information. The goal is to make the victim believe that the web page is legitimately asking for a second factor of authentication or other sensitive information (as illustrated above). In fact, the victim will notice no suspicious signs (e.g., invalid SSL - Secure Sockets Layer certificate or different URL - Uniform Resource Locator) because the page is modified “on the fly” right before being displayed, directly on the local machine.

WebInjects have evolved over time, starting from simple phishing-like key-loggers to offering automatic transfer systems (ATS) and two-factor authentication bypass, together with mobile components and web control panels to manage money and fraudulent transfers [8]. Custom WebInjects can be also purchased for as little as a few tens of USD. Furthermore, cybercriminals offer paid support and customization, or sell advanced configuration files that the end-users can include in their custom builds.

Since banks implemented two-factor authentication using One Time Passwords (OTPs) sent by SMS, in the last years most of the banking Trojans toolkits included a mobile component. This mobile component works in pairs with the PC versions and can access all the information in the user's phone, including SMS, and send it to its C&C server. This attack scheme is also known as “Man in the Mobile” (MitMo). Once the victim's PC is infected, when the victim visits his online banking website the Trojan steals his credentials and inserts a message in the web page that invites the user to download and install a new mobile application to be able to access his account from his mobile phone. This step is usually performed inserting in the web page a QR code that points to the malicious application's download. When the victim downloads and installs the mobile malware, her phone is compromised. The mobile malware can now intercept all the SMS, silently avoid the system notification and remove them after they have been sent to the aggressor.

2.1.2 Crypto-Ransomware

Crypto-Ransomware is a class of malware that encrypts valuable files found on the victim's machine and asks for a ransom to release the decryption key(s) needed to recover the plaintext files.

Figure 2 Example of a ransom notes (Locky, CTBLocker)

Quite interestingly, this class of malware was predicted with uncanny accuracy 20 years ago, in a research paper by Young et al. [9]. The requested ransom payment is typically in the order of a few hundred US dollars [10] (or equivalent in crypto or otherwise untraceable currency) [11]. Clearly, the success of these attacks depends on whether most of the victims agree to pay (e.g., because of the fear of losing their data). Unfortunately, according to a thorough survey dated November 2015 [12], about 50% of ransomware victims surrender to the extortion scheme, resulting in millions of dollars of illicit revenue. In the first three months of 2016, according to a recent analysis [13], more than 209 million US$ in ransomware payments were made in the US alone.

From a technical viewpoint, the sophistication of ransomware families has increased over time. While first-generation ransomware families were cryptographically weak, succeeding families adopted more elaborated principles such as encrypting each file with a unique symmetric key protected by public-key cryptography. Consequently, the chances of a successfully recovery (without paying the ransom) have drastically decreased.

2.2 The importance of cryptocurrencies for financially motivated malware

A vital element of any financially motivated cybercrime is liquidation, or extraction of financial value from the criminal activity. Cryptocurrencies have become a cornerstone of digital crime, with many newer currencies offering features such as:

- Privacy
- Claims of anonymity or pseudonymity
- Claims of being untraceable

Cryptocurrencies, such as Bitcoin, have played a fundamental role in the success of Cryptolocker and its ilk. Traditionally, payment methods such as UKash and PaySafeCard have been used as a way for individuals to pay ransoms using fiat currency over a digital medium. However, the pseudonymity and ease of transfer provided by Bitcoin has led to a decline in such methods.

Creating wallets is trivial, and wallets that are not used in conversion to or from fiat currency (using Bitcoin to purchase GBP for example) aren't easily tied to identifying information. The only common identifier for most cryptocurrency wallets is their public key, which is used as a deposit address. This information is provided when demanding a ransom, and multiple addresses are used in many implementations.

Only leaving one step between ransom and cashing out would likely compromise the ransomware operator, as it would be possible to trace the fiat-facing transaction. As a result, a sophisticated laundering process is employed, with near universal steps taken to reduce the opportunity for law enforcement agencies to successfully intercept the entirety of the proceeds of this criminal activity.

After receiving a set amount of currency, these wallets transfer their balance through a series of wallets which commonly only have 2 transactions each. The first of these is the receipt of bitcoin and the second is to split the incoming value between two other addresses. This is known as a 'peeling chain' and is common to Bitcoin. Observations of the scale and near-identical timing of such transactions indicates that the laundering of Bitcoin in this manner is automated. This is known as mixing or tumbling, and is an enterprise that demands a 2.5% fee from the laundered balance. This represents an operational cost, as does the cost of transfer incurred when any bitcoin transaction is undertaken.

Sean Sullivan suggests that Bitcoin friction may be ransomware's only constraint. This friction is generated by individuals who are willing to pay but unable to access bitcoin, either due to some constraint (such as lacking sufficient personal identifying information to make the initial transaction from an exchange), or inability to acquire the currency in the (sometimes short) allotted time. The volatility of Bitcoin is another factor, as Bitcoin can rapidly increase or decrease in value on an hourly basis. This puts pressure on ransomware operators to constantly adjust prices and provide short periods of time between initial infection and ransom deadlines. These factors reduce the total potential profit of the enterprise, regardless of the ransom strategy itself; they represent technical challenges in the medium of exchange.

Bitcoin has proven popular because it is fungible and it is easily converted into a fiat currency value. However, it is traceable, and this may imply a coming change in how criminals interact with cryptocurrencies. In an increasingly competitive environment, ransomware must compete technologically (infection rate and exploitation of vulnerabilities) and fiscally (extraction of cash value without being caught/denied payment). Currencies that offer increased anonymity and protection from potential tracking by LEAs will be attractive. If large criminally-focused enterprises such as AlphaBay back currencies by allowing it in their marketplace, then this will stimulate use and strategies involving more effective means of avoiding financial paper trails during the critical liquidation process.

Identification of the market activity and capacity of such alternative currencies will give LEAs an idea of how attractive each currency is as an alternative to bitcoin, and their likelihood of being involved in major ransomware crimes. Newer currencies have a smaller pool of currency, and some have a low market footprint by design, limited to hundreds of thousands of dollars of maximum potential currency in circulation.

2.4 State of the art

Law Enforcement Agencies, in an effort to fight against new digital crime and collect relevant digital evidence, are incorporating computer forensics techniques into their infrastructure in order to stop the fast growth of this type of crime.

The vertiginous change of technology has converted these tasks into a constant race between the criminals and the LEA's. That is why the use of new forensic techniques will permit LEA's to prevent crime and also, catch all the criminals behind.

Recent cases of child pornography and ransomware extortion, both in Portugal and Spain [1-3], as well as in Europe have increased the interest by the LEA's to improve their current forensic techniques to decrease the presence of these cases in their respective countries.

2.4.1 Existing tools and techniques for LEA and for the general market

LEAs current approaches to counter and prevent cybercrime (especially financially driven) can be categorised into three main topics: strategy, forensic expertise and operations:

- Strategy is mostly linked to continuous efforts to empower human resources, to increase the institutional capacity building to fight back against cybercrime, and to further promote networking and cooperation on a European and international scale.
- Forensic expertise and digital forensic are becoming crucial for the LEAs during all their activities in order to investigate and prosecute cybercrime, but this is a complex area which require high skills and competences.
- Operations mostly relate to cyber intelligence and new intelligence disciplines, with training and education being a fundamental resource for LEAs.

2.4.2 Partner research tools

Screenshot-Based Classification - Our partners from USAAR are developing a system to analyze ransomware samples. Specifically, this tool will be able to identify the ransomware family/variant from a screenshot of the ransom note left during the infection. By using Optical Character Recognition (OCR) the tool will identify the text showed on the victim's computer, store it, and compare it with a set of samples in the RAMSES database to recognize the ransomware attack. LEAs will be able to upload a screenshot of the infected machine and visualize results about the ransomware variant and the campaign behind the attack.

Tool for video and image analysis - This tool will be capable of linking a set of images or videos to a particular device (digital camera, smartphone) will be developed and a model will be made. These tools will also be capable of, using a different number of digital image forensic algorithms in order to detect image and video manipulation.

Steganography detection in multimedia - This tool will identify the use of steganography in various kinds of malware. The focus will be on banking Trojans and ransomware. Steganography will be identified through analysis of signatures left by the algorithms used. This allows for faster analysis of images to determine whether they contain hidden content. The purpose of this tool is not to identify the content hidden in images or video, but to determine if there is any hidden content. This will apply to image, video, voice over IP, audio and other multimedia. The focus, however, will be on images and video, as they are the most likely avenues for the kind of steganography supported communication we expect the address in this project.

Sandnet - Sandnet is the malware analysis system used by USAAR to collect information about current malware trends. The malware feeds of SandNet deliver thousands of new malware samples every day which are then executed and monitored in virtual machines. The monitoring process collects analysis data that can be used by the other tools to cluster and classify malware. The analysis data includes but is not limited to: Network traffic, File system activity, Screenshots of the virtual machine display, Memory dumps, and API calls.

Clustering - The analysis data can be used to cluster malware families to gain insights into current malware trends and campaigns. A reliable way of doing this is to use the network traffic and build network-based signatures to identify malware families which use command and control (C&C) based communication [14]. For malware types which have a characteristic visual appearance as for example in the case of ransomware, the screenshot can be used to perform the clustering [15].

2.5 References

[1] Goncharov M., Russian underground 101. Trend Micro Inc. Research Paper, 2012

[2] Lindorfer M., Di Federico A., Milani Comparetti P., Maggi F., Zanero S., Lines of Malicious Code: Insights into the Malicious Software Industry. In Annual Computer Security Applications Conference, 2012.

[3] Grier C., Ballard L., Caballero J., Chachra N., Dietrich C. J., Levchenko K., Mavrommatis P., McCoy D., Nappa A., Pitsillidis A., et al. Manufacturing compromise: the emergence of exploit-as-a-service. In Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012

[4] Doherty S., Krysiuk P., Wueest C., The state of financial Trojans 2013. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_state_of_financial_Trojans_2013.pdf

[5] https://zeustracker.abuse.ch/statistic.php

[6] http://www.team-cymru.org/Services/MHR/

[7] Wueest C., The state of financial Trojans 2014. Symantec, 2014

[8] Boutin J. I., The evolution of webinjects. ESET, https://www.virusbtn.com/pdf/conference/vb2014/VB2014-Boutin.pdf , 2014

[9] Young A., Yung M., Cryptovirology: Extortion-based security threats and countermeasures, 1996 IEEE Symposium on Security and Privacy, pp. 129--140

[10] Savage K., Coogan P., Lau H., The evolution of ransomware. Symantec, 2015

[11] Spagnuolo M., Maggi F., Zanero S., BitIodine: Extracting Intelligence from the Bitcoin Network. 18th International Conference on Financial Cryptography and Data Security, pp. 457-468, Christ Church, Barbados, March 3-7, 2014

[12] Arsene L., Gheorghe A., Ransomware. A Victim's Perspective. Bitdefender, 2016. http://www.bitdefender.com/media/materials/white-papers/en/Bitdefender_Ransomware_A_Victim_Perspective.pdf

[13] Trend Micro. Ransomware Bill Seeks to Curb the Extortion Malware Epidemic. April 2016. http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ransomware-bill-curb-the-extortion-malware-epidemic

[14] C. J. Dietrich, C. Rossow, N. Pohlmann, CoCoSpot: Clustering and Recognizing Botnet Command and Control Channels using Traffic Analysis. Available at http://www.christian-rossow.de/publications/cocospot2012.pdf (Last consultation: 22 May 2017).

[15] C. J. Dietrich, C. Rossow, N. Pohlmann, Exploiting Visual Appearance to Cluster and Detect Rogue Software. Available at http://www.christian-rossow.de/publications/visualmalware-SAC2013.pdf (Last consultation: 22 May 2017).

3 Requirements and specifications

The RAMSES end-users are police forces from Belgium, Germany (through BayFHVR, Bavarian Police College), Spain and Portugal and most of them are specialised units dealing with cybercrime. Their success depends on the availability of both relevant information about the specific case under investigation and on a sound knowledge of the phenomenon/criminal category the specific case belongs to. Regarding financially motivated malware (specifically to banking Trojans and ransomware), the domain is immense and unpredictable, and the quantity/quality of available information are massive and ever changing. This determines that the knowledge-generation process is complex and time-consuming, and it needs specific skills and a continuous upgrade. Therefore, it can't be handled manually to be effective and efficient.

The RAMSES Project aims to contribute at improving the law enforcement success rate by designing and developing a holistic, intelligent, scalable and modular software platform for Law Enforcement Agencies (LEAs) to facilitate digital forensic investigations. The system should extract, analyze, link and interpret information extracted from the Internet (surface and deep web) and related with financially-motivated malware. In order to develop the specific functionalities of the platform around a set of relevant actual needs of the law enforcement agencies involved in the investigation process, a requirements analysis process was developed, based on the mutual cooperation and a teamwork between the LEAs and the technical partners, described in the deliverables *D.2.2 Report on relevant scenarios* [1] and *D.2.3 Report on user requirement for the design of LEAs tools* [2].

3.3 Requirement elicitation from LEAs

In order to elicit requirements from the LEAs, we leveraged the different meetings that we attended together with the other partners of the consortium. Specifically, we presented to the LEAs our initial idea and design of the analysis systems, describing its inputs, outputs, and functionalities. During these meetings the LEAs provided us feedback and specific requirements needed for their activity. We also exchanged e-mails to update LEAs about the progress of our design and receive approval for our choices.

3.3.1 LEAs main current practices

During the first phase of the RAMSES project implementation, they have shared some inputs about their mission, activities, but also problems and needs, as well as expectations. In particular, they provided information about manual searching, automated searching, databases, examination of electronic evidence and cooperation with other LEAs units. These are the main activities that they perform, and that require technology assistance.

- **Manual searching**: It is currently used mostly to deal with Open-source Intelligence (OSINT), to do cross-check activities, to process metadata. The most relevant problems are related to the lack of skills and capacity, as well as human resources. The expectations seem to be related mostly to the possibility to automatize the processes and to include also the sources and information available in the Deep Web and the Dark-Nets.

- **Automated searching**: It is currently used mostly to do the crawling of some specific sources (such as markets, forums and social media). The most relevant problems are related to the lack of human resources as well as to the difficulties determined by the storing and the analysis of the data collected. The expectations seem to be related mostly to the possibility to analyze the sources and information available in the Deep Web and Dark Nets.

- **Databases**: LEAs already have some databases used to store for example "hashes, sources, target sectors, information from crawlers, honeypots and malware analysis results". The most relevant problems are related to the manual export, which is not adequately fast and efficient. The expectations seem to be related mostly to the possibility to have front-end functionalities that allow to perform searches and visualise relations, and cross-check technical information with criminal records and intelligence databases.

- **Examination of electronic evidence:** It is used for the identification of malware especially from the victims' computers or thanks to contributions from several sectors and CERTs; The most relevant problems are related to the lack of skills and capacity. As computers and smartphones require different approaches and tools, they generate heterogeneous expected results. Second, the difficulty to reach the victims promptly and efficiently, and to identify and extract malware samples and vectors from their devices. The expectations seem to be related mostly to the availability of identification tools, intended also as tools to automatize the analysis of the victim's computer remotely with the screenshot of the ransomware message received, perform malware triage (how to identify the various types of security incidents by understanding how attacks unfold, and how to effectively respond before they get out of hand). Finally, perform cross-checks with Indicator of compromise (IOCs).

- **Cooperation with other LEAs units:** Existing initiatives are limited only to some stakeholders and they often have security problems. Confidentiality is also a key restriction. Expectations seem to be related mostly to the enhancement in the mechanisms to exchange information based on both technical and non-technical solutions.

3.3.2 Key results

From the discussion and interviews that we conducted with the LEAs, we obtained some requirements that specifically affect our task, the design and development of analysis systems.

In summary, these are the requirements of the LEAs:

- Create an innovative software solution using Open Source technologies to support mostly the early stages of the investigation process;
- high efficiency in accessing relevant data sources and retrieving information significant for forensic investigation thanks to innovative Big Data technologies.
- A secure platform to avoid attacks to the application and to keep the privacy, confidentiality, integrity, and availability of the data.
- Data-gathering for investigative operations;
- Digital forensic;
- Understanding of malware samples, with specific regards to:
    - Early-gathering of malware samples;
    - Early-detection of malware samples;
    - Malware family identification;
- Analysis of the malware samples collected to extract useful information for the investigations (e.g., endpoints);
- Fast-response to victims (intended as for example tools and functionalities – e.g. downloadable by the victim(s) from the project website - to promptly activate an automatic "analysis" of the targeted machine(s) to collect key information about the virus).
- Analyzing and profiling the Bitcoin network. Specifically:
    - Group Similar Payments
    - Associate Entity to Payment

- Provide interfaces to perform queries and visualize results

3.4 Requirement analysis: cryptocurrencies other than Bitcoin[1]

Recent entrants to the cryptocurrency market include a handful of new currencies that claim to provide privacy, anonymity, and untraceability as core features. Identification of the market activity and capacity of such alternative currencies will give LEAs an idea of how attractive each currency is as an alternative to bitcoin, and their likelihood of being involved in major ransomware crimes. Some examples include XDN, Zcash, and Monero. These three currencies represent some of the cutting-edge contributions to the cryptocurrency domain, with elements that by fortune or design favour dark web dealers and cybercriminals. Developed with privacy in mind, all three of these currencies purport to allow anonymity and mostly importantly untraceable transactions. XDN is an open currency with community tools provided for ongoing work. Zcash is a closed standard, developed around the concept of zero-knowledge proofs, a particularly interesting attribute with effectively allows transactions to be made between two legitimate users with no actionable knowledge of each other ever being communicated. Monero is not based on bitcoins code in any way. It is a common criticism that many cryptocurrencies that claim improved security are built on a potentially flawed premise, as Bitcoin is not engineered for true anonymity, merely privacy at the blockchain level.

Bitcoin has proven popular because it is fungible, it is easily converted into a fiat currency value. However, it is traceable, and this may imply a coming change in how criminals interact with cryptocurrencies. In an increasingly competitive environment, ransomware must compete technologically (infection rate and exploitation of vulnerabilities) and fiscally (extraction of cash value without being caught/denied payment). Currencies that offer increased anonymity and protection from potential tracking by LEAs will be attractive, even more so now that currencies such as Monero are accepted by AlphaBay. If large, criminally-focused enterprises such as AlphaBay back currencies by allowing it in their marketplace, this will stimulate use and strategies involving more effective means of avoiding financial paper trails during the critical liquidation process.

3.4.1 Alternative cryptocurrency assessment methodology

To provide an overview of the emerging role of altcoins in cybercriminal activities, three key areas were investigated:

- The technical attributes of altcoins conducive to anonymity, privacy and cost-reduction
- Their current market capitalisation, adoption by known DNM and reports of malware demanding payment in altcoins (alongside Bitcoin or exclusively)
- Social media and crypto-currency community coverage, related activity and bleed-over into mainstream media channels

By collecting data, reports and statements relating to these three areas, it is possible to define altcoins in terms of their utility, knowledge associated with them and current penetration in both market and media sectors. This latter characteristic, though the least important in terms of identifying traits that allow an altcoin to circumvent current data forensics techniques, is vital from the perspective of the acknowledgement of those attributes by the cryptocurrency community. This includes cybercriminal actors and members of the public: differentiating privacy-rights advocacy from criminal intent is neither in the scope of this work, nor does it add value to the analysis of these currencies from the perspective of digital forensics. We make no comment as to the motivations of users, but directly relate the capabilities of each currency to examples of malicious use.

[1] University of Kent contribution - Author: Darren Hurley-Smith

The following methods were employed to collect this data:

- Literature review: scientific literature pertaining to zero-knowledge proofs, ring signatures and other privacy features has been used to provide a scientific grounding for our observations regarding the likely effectiveness of existing blockchain analysis techniques and online-observation of users of a given altcoin.
- Altcoin whitepapers: reading the whitepapers providing by coin developers allows a list of their promised features to be compared against their current achievements. Tis can be combined with community and literary feedback, to comment more accurately on what a given altcoin development team wishes to achieve and where they are currently.
- Social media (Reddit/Twitter), Market (GDAX, Coinbase, Worldcoin Index), and mainstream media coverage of coins can be used in combination to comment on the capitalisation, usage and current public perception of altcoins. Cryptocurrency remains the province of a very specialised minority, but it is becoming more visible, especially with the number of high profile attacks reported year on year.
- Hands on experience: Interacting with the wallet clients and blockchain explorers for Monero, Ethereum and Dash provides a narrow but direct insight into the usability of such currencies and the technical skills required to safely and securely operate them.
    - This testing was performed offline in the case of wallet clients
    - No currency used, wallet set up and initialisation was performed on an air-gapped machine and no currency was sourced, received or exchanged – all balances start, remain and end at 0.
    - Blockchain interaction/observation tools were used online. XMRchain and Etherchain were used to highlight differences between an opaque and transparent chain.

3.4.2 Key results

From our reading, discussions and analyses, we identified the key attributes of 5 prominent altcoins. These currencies incorporate or plan to incorporate privacy features, and three of them are now in the top ten currencies (by market capitalisation – a good indication of their popularity).

In summary, our findings include:

- Monero, ZCash and Dash all possess blockchain analysis-defying features
- Monero and Dash are in the top ten cryptocurrencies by market capitalisation
    - Dash is most popular due to its instant send functionality, less than 28% of users make use of its privacy features
    - Not using privacy features leaves users exposed to the usual blockchain analysis techniques that are useful against Bitcoin and Ethereum-like chains
    - Monero is private by design, there is no transparent node. However, it should be observed that Monero is no replacement for good personal operational security (opsec): it makes personal opsec the difference between privacy and observation but makes it impossible to derive information about transactions that remain undisclosed
    - Monero ensures fungibility by masking the input-output relationships in transactions, by grouping them with other transactions and obfuscating the links between them. In this way, the link between endpoint users is masked.
    - ZCash achieves a similar outcome, by using zero-knowledge proofs to provide privacy and non-repudiation features to transactions.
- Ethereum is the 2nd most capitalised cryptocurrency

    - Ethereum is a token-backed network, which allows users to develop distributed applications and create smart-contracts. These are digital agreements that are verified by the blockchain, allowing a vast number of contractual tasks to be processed, validated and enforced based on the high-level of security offered by the blockchain. Autonomous contract resolution is an example feature.
    - Ethereum is a good example of current generation cryptocurrencies: they are no longer merely stores of value of cash alternatives. They must offer functionality.
    - Ethereum is not private, it operates a transparent chain. However, zK-Snark implementation is being discussed and the foundations will be laid later this year with the Metropolis hard-fork.
- Cybercriminal activity using Ethereum and Monero has been reported, though rarely does it circulate to the mainstream media
    - This indicates that currently, Ethereum and Monero are low-adoption. It may also indicate that (at least for Monero) they are effective at avoiding any personal data leakage or other information that would lead to arrests being made (which would increase the media footprint associated with the currency).
    - Ethereum and Monero are both accepted on several DNM, notably Alphabay and Hanza prior to their shutdown. Ethereum offers lower transaction fees, but the benefits of Monero for illicit activity are obvious.
    - Two malware strains have been associated with Monero: the Wannacry operators laundered their Bitcoin through Monero and the Kirk ransomware demanded payment in Monero. Neither have proven particularly effective, though Wannacry received significant media attention. The exfiltration of money, however, was not paid much attention outside technical or cryptocurrency community venues.

Our report concludes that these currencies are largely held back by a lack of acknowledgement and adoption by the wider public. The cryptocurrency community, in terms of the technically proficient and daily participating members, know of and are likely invested in these altcoins in some way. There have even been charity drives and a fully funded PhD that have been driven by the Monero community using their chosen currency, so there's a high degree of utilisation by those involved with the currency at present. However, as with Bitcoin, the public interest is the deciding factor in use and adoption. Dash is benefitting from a wave of increased attention in developing economies, where preservation of value and quick, cheap international transactions are beneficial when local economies are volatile. However, this is limited to its fast transaction features: privacy is a priority for less than 28% of its users.

It remains difficult to safely configure and set up Monero, Dash and ZCash wallets that provide their full features. Dash wallets are easy to configure, especially in multi-wallet applications, but do not offer the privacy features that are of likely to appeal to cybercriminals. The extra effort of explaining how to configure these wallets and acquire the currency to pay a ransom is an additional cost to ransomware operators.

Privacy-focused altcoins will play and increasing role in cybercriminal activity. They provide the means to defy most effective digital forensic techniques. On-chain and online forensics are unlikely to be effective against Monero and ZCash unless advances are made in areas specific to zero-knowledge proofs and ring-signature derivation. Offline techniques will remain viable, and privacy-centric currencies are still only as good as the opsec of the user. Bitcoin will likely remain dominant due to its proliferation and public familiarity with its existence, but it will be important to stay aware of mainstream media coverage of

cryptocurrencies, to gauge their proliferation amongst the potential users, operators and victims of cybercrime.

4. System design

This module will be composed by two different tools: BitIodine and Prometheus. Each of these tools allow different and automated types of forensic analysis.

BitIodine is a framework that is focused on the automatic extraction of intelligence from the bitcoin network in order to classify and visualize this information, which in a manually way wouldn't be possible to carry out.

Prometheus is a tool that is focused on the automatic identification and analysis of banking trojans that modify the DOM in order to steal banking information from a victim.

4.1 Summary of specifications

Here we summarize the specifications drawn from Section 3. Specifically, we describe the requirements of both the analysis systems we are developing.

Prometheus: Memory Forensics for Banking Trojan Analysis and Detection

- Malware Detection: Verify if a victim's machine is infected by a banking trojans.
- Family Identification: Recognize the family and/or variant of the trojan that infected the victim's machine.
- Information Gathering: Extract useful information from the infected machine for the investigation, such as the endpoints (e.g., C&C server address) that are contacted by the sample.
- Fast-response to Victims: Provide an interface to easily activate an automatic "analysis" of the targeted machine(s) to collect key information about the infection.

BitIodine: Extracting Intelligence from the Bitcoin Network

- Analyzing and profiling the Bitcoin network
- Group Similar Payments: Cluster transactions that are likely to belong to a same user or group of users
- Associate Entity to Payment: Associate an entity to the identified clusters.
- Interface: Provide an easy-to-use interface to perform queries and visualize results.

4.2 Overall system design

Here we describe the overall design of the two systems and their components, inputs and outputs, data sources and data flow.

4.2.1 Prometheus: Memory Forensics for Banking Trojan Analysis and Detection

Prometheus is an automatic system that is able to analyze banking trojans that base their attack technique on DOM modifications. From a technical point of view such malware is equipped with a functionality, called WebInject, that exploits API hooking techniques to intercept all sensitive data in a browser context and modify web pages on infected hosts. This type of malware is able to intercept data after being decrypted, thus essentially nullifying the added security by secure transport protocols such as HTTPS. The goal of this malware is to make the victim believe that the web page is legitimately asking for a second factor of

authentication or other sensitive information and the victim will notice no suspicious signs because the page is modified "on the fly" right before being displayed, directly on the local machine. New families and new versions of banking trojans are frequently released and each specific trojan can be customized and obfuscated, generating new, distinct executables. In addition, the custom configuration files are encrypted and embedded in the final executable. For these reasons, manually analyzing all the samples is not scalable. Thus, automatic mechanisms to extract valuable information from encrypted configuration files or for analyzing activity of an infected machine are needed.

Webinject-based trojans leave artifacts of the injection behaviour in the infected machine's memory, e.g., list of targets URLs. Prometheus, leveraging memory forensics techniques, is able to inspect memory and extract these artifacts that can be used as indicators of compromise. Specifically, our approach is able to recover such artifacts from the memory of an infected machine by inspecting the memory of the target browsers, and searching for strings tokens associated to the definition of regular expressions, part of any WebInject configuration file.

As is shown in Figure 3, Prometheus will provide an HTTP API in order to work with RAMSES Platform. The API will take as input a (complete or partial) memory dump obtained from the infected machine and will produce in output the trojan family detected on such data. When possible it will also output info about the malware, such as URL address of the C&C server and list of the URLs targeted by the malware. To extract the target URLs and regular expressions, we developed a Volatility plugin based on YARA. This plugin scans the memory dump, looking for all the strings that match a well-designed YARA rule. In particular, since we observed that the URLs and the regular expressions are loaded in the browser's memory, the plugin inspects only its address space. We defined a regular expression that matches the pseudo-URL format of the WebInject target URLs (e.g., domain.com/ibank/transfers/* , bank.com/login.php*). Moreover, our YARA rule filters the matched strings that are close to each other. In fact, since we noticed that the WebInject targets are allocated sequentially, we leverage this fact to exclude all the matching strings that are not WebInject targets.

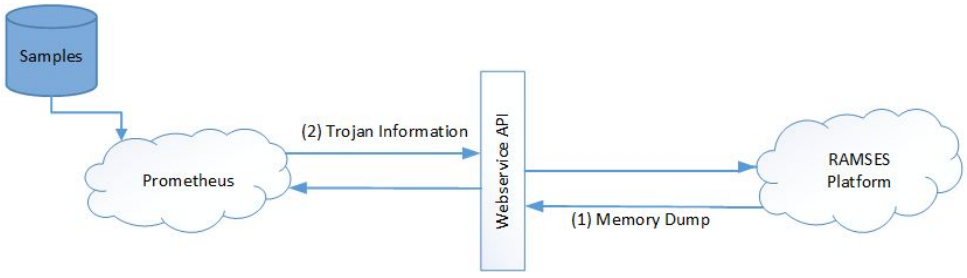

Figure 3: **Prometheus Communication API**.

USAGE

The expert end users (LEA or academics) should take a memory dump of the infected machine. To avoid privacy issues, law enforcement agencies will remove in a semi-automatic fashion (i.e., through the application of heuristics and manual inspection) the sensitive data from the memory dump leaving only the memory regions that may contain infection artifacts. After removing the sensitive data, they will upload the partial memory dump to the RAMSES platform to start the analysis. When Prometheus finishes the analysis of such dump, the end user can obtain and inspect the results of the analysis.

4.2.2 BitIodine: Extracting Intelligence from the Bitcoin Network

BitIodine is a modular framework which parses the blockchain, clusters address that are likely to belong to a same user or group of users, classifies such users and labels them, and finally visualizes complex information extracted from the Bitcoin network. It can label users in a semi-automatic fashion with information on their identity and actions which is automatically scraped from openly available information sources.

Bitcoin is a decentralized monetary system based on an open-source protocol and peer-to-peer network of participants that validates and certifies all transactions. Each node of the network must store the entire history of every transaction ever happened, called *blockchain*. Thanks to the fact that all Bitcoin transactions are public and transparent, anyone can reconstruct the entire flow from address to address. By analyzing the blockchain and correlating it with publicly available metadata, it is possible to find addresses used for illegal activities (e.g., for gambling, mining, or for scans). These addresses can also be algorithmically grouped in clusters that correspond with entities that control them, but do not necessarily own them.

Figure 4 describes in a simplified way the building blocks of BitIodine and the interactions between different modules.

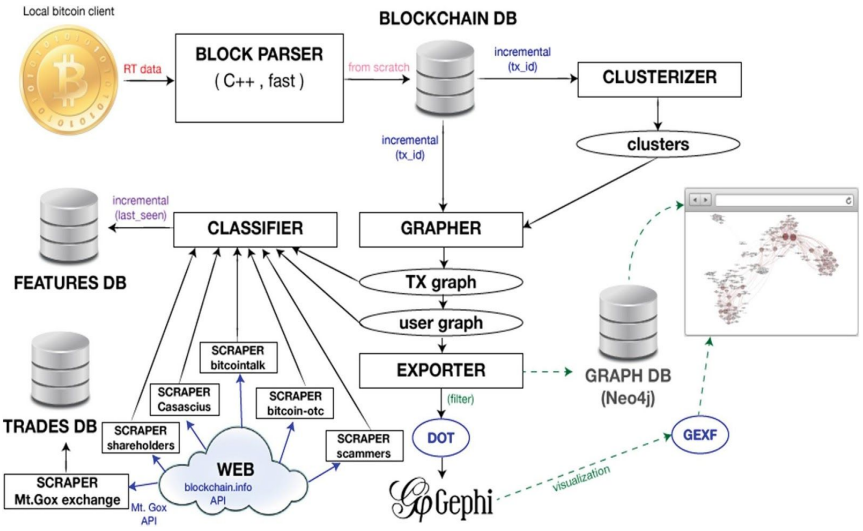

Figure 4: **BitIodine components**

The **Block Parser** reads blocks and transactions from the local bitcoin folder populated by the official bitcoin client and exports the blockchain data to the blockchain DB, which uses a custom relational schema. This allows for a fast updating of data from the Bitcoin network.

The goal of the **Clusterizer** is to find groups of addresses that belong to the same user. It incrementally reads the blockchain DB and generates clusters of addresses using different heuristics. Clusters are stored in cluster files.

A set of **Scrapers** crawl the web for Bitcoin addresses to be associated to real users, automatically collecting, generating and updating lists of:

- usernames on platforms, namely Bitcoin Talk forum and Bitcoin-OTC marketplace (from forum signatures and databases).
- physical coins created by Casascius (https://www.casascius.com) along with their Bitcoin value and status (opened, untouched).
- known scammers, by automatically identifying users that have significant negative feedback on the

Bitcoin-OTC and Bitcoin Talk trust system.

- shareholders in stock exchanges (currently limited to BitFunder).

Additional lists can be built with a semi-automatic approach which requires user intervention. In particular, by downloading tagged data from https://blockchain.info/tags, the tool helps users build lists of gambling addresses, online wallet addresses, mining pool addresses and addresses which were subject to seizure by law enforcement authorities. The user can verify tags and decide to put the most relevant ones in the correct lists. Finally, a scraper uses Mt. Gox trading APIs to get historical data about trades of Bitcoin for US dollars, and saves them in a database called trades DB. This module is useful to detect interesting flows of coins that enter and exit the Bitcoin economy. The interface is easily expandable, and adding scrapers for new services and websites is easy.

The **Grapher** incrementally reads the blockchain DB and the cluster files to generate, respectively, a transaction graph and a user graph. In a transaction graph, addresses are nodes and single transactions are edges. The Grapher has several applications (e.g., finding successors and predecessors of an address). In a user graph, users (i.e., clusters) are represented as nodes, and the aggregate transactions between them are represented as edges.

The **Classifier** reads the transaction graph and the user graph generated by the grapher, and proceeds to automatically label both single addresses and clusters with specific annotations. Examples of labels are Bitcoin Talk and Bitcoin-OTC usernames, the ratio of transactions coming from direct or pooled mining, to/from gambling sites, exchanges, web wallets, other known BitcoinTalk or Bitcoin-OTC users, freebies and donation addresses. There are also boolean flags, such as one-time address, disposable, old, new, empty, scammer, miner, shareholder, FBI, Silk Road, killer and malware. Classification can take place globally on the whole blockchain, or selectively on a list of specified addresses and clusters of interest. The results are stored in a database and can be updated incrementally.

The **Exporter** allows to export and filter (portions of) the transaction graph and the user graph in several formats, and support manual analysis by finding simple paths (i.e., paths with no repeated nodes) on such graphs. More precisely, it can export transactions that occurred inside a cluster, or that originated from a cluster. It can also find either the shortest, or all the simple paths from an address to another address, from an address to a cluster, from a cluster to an address, or between two clusters. Moreover, it can find all simple paths originating from an address or a cluster (i.e., the subgraph of successors), or to reverse such search, by identifying the subgraph of predecessors of an address or cluster. Subgraphs of successors or predecessors can be useful, for instance, in taint analysis, and can assist manual investigation of mixing services.

The main characteristics presents in RAMSES Platform, by BitIodine, will be:

- Export and filter (portions of) the transaction graph and the user graph.
- Find simple paths (i.e., paths with no repeated nodes) on such graphs.
- Export transactions that occurred inside a cluster, or that originated from a cluster, find either the shortest, or all the simple paths from an address to another address, from an address to a cluster, from a cluster to an address, or between two clusters.

BitIodine will provide a set of APIs to access the data produced. As input BitIodine takes blocks and transactions from the blockchain data. Our APIs will allow to access data in BitIodine's graph DB and perform query on such data.

USAGE

BitIodine supports manual investigation by finding (reverse) paths between two addresses or a user and an address. By accessing the platform, authorized users (LEA and academics) can query the results that BitIodine automatically produce to obtain useful information for the investigations.

5. Implementation

Here we describe the basic technologies used, the implementation choices followed, the parameters that each tool will require, and the results that will be produced. Table 1 summarize the design of the proposed tool.

Service name	Module for the forensics analysis of malware payments and infected machine memory.
Service Description	This module will be composed by two different tools: BitIodine and Prometheus. Each of these tools allow different and automated types of forensic analysis. BitIodine is a framework that is focused on the automatic extraction of intelligence from the bitcoin network in order to classify and visualize this information, which would not be possible to carry out in a manual fashion. Prometheus is a tool that is focused on the automatic identification and analysis of banking trojans that modify the DOM in order to steal banking information from a victim.
Other comments	When using Prometheus, end users (LEA or academics) should take a (complete o partial) memory dump of the infected machine and upload it on the platform. When Prometheus finishes the analysis of such dump, the end user can obtain and inspect the results of the analysis. BitIodine supports manual investigation by finding (reverse) paths between two addresses or a user and an address. By accessing the platform, authorized users (LEA and academics) can query the results that BitIodine automatically produce to obtain useful information for the investigations.

Table 1 - Service specification

5.1 Prometheus

Prometheus is entirely developed in Python. The implementation of the memory forensic module is based on Volatility, a popular, open-source memory forensics framework. Specifically, we developed a volatility plugin that implements our approach. Such plugin also uses YARA, a pattern matching tool that allows to define complex rule to search and match certain content. We use YARA to define custom rules that allow us to identify artifacts left in memory by trojans during their malicious activity. The most important role of the memory analysis is extracting from the memory dump the URLs and regular expression that identify the WebInject targets. Our plugin extracts the strings the satisfies the designed YARA rules and that are allocated sequentially in the address space of the browser (e.g., Internet Explorer).
Prometheus also present a web interface through which it is possible to submit the analysis of memory dumps. The web interface is implemented in Python using the flask framework, which allow to provide both restful APIs and a graphic interface.

Name	Type	Description
Analysis ID	int	A unique identifier for the analysis
Sample Hash	text	Sample Hash (SHA256)
Sample	blob	Sample Binary
Memory dump	blob	Partial Memory dump
C&C URL	text	C&C URL extracted from the analysis
Family	text	Identified family
Encryption keys	List of strings	Encryption keys extracted during the analysis
WebInject Targets	List of strings	WebInject targets (in the format of URLs of regexp) extracted during the analysis
Other	text	Other raw artifacts found in the memory during the analysis

Table 2 - Prometheus parameters and results

Prometheus API will take as input a memory dump obtained from the infected machine and will produce in output the trojan family detected on such data. When possible it will also output info about the malware, such as URL address of the C&C server and list of the URLs targeted by the malware.

5.2 BioIodine

We foresee BitIodine dealing with several gigabytes of data and graphs with millions of nodes and tenths of millions of edges. For this reason, we used Python 3.3.3rc1 for every module, except the **Block Parser**, which is written in C++ for performance reasons. The block parser is a modified version of the blockparser tool by *znort987* (http://github.com/znort987/blockparser) , to which we added several custom callbacks: our modified version is highly efficient in exporting all addresses on the network, in performing taint analysis on an address, and in exporting to SQLite.

We opted for the use of embedded SQLite databases for storing the blockchain and the features database because it is a zero-configuration, server-less, embedded, stable and compact cross-platform solution. We do not need concurrency while writing to database files, so the only possible disadvantage does not affect its use in BitIodine. In designing the custom database schema for BitIodine we had to find a good balance between size and performance, weighing the use of indexes.

The **Clusterizer** is designed to be incremental, and it is also possible to pause the generation of clusters at any time, and resume it from where it stopped. Internally, graphs are handled by NetworkX, which objects can be serialized and written to a file with ease, and in-memory querying for successors and predecessors of nodes is efficient. Is it also possible to embed an arbitrary number of additional data labels to nodes and edges (e.g., we added transaction hashes).

The **Exporter** supports several output formats, allowing easy pipelining with visualization software or graph databases.

Name	Type	Description
Query	text	Query String containing info about transactions and blockchain
Transactions	List of objects	Transactions selected from the query
Path	List of objects	Path between addresses or clusters
Graphs	List of nodes/edges	Portion of the user graph
Others	text	Other info extracted from the query

Table 3 - BitIodine parameters and results

The BitIodine API will take blocks and transactions from the blockchain as input and allow to access and to perform queries on BitIodine's data. In particular, it will allow to export and filter (portions of) the transaction and the user graph, find simple paths (i.e., paths with no repeated nodes) on such graphs, export transactions that occurred inside a cluster, or that originated from a cluster, find either the shortest, or all the simple paths from an address to another address, from an address to a cluster, from a cluster to an address, or between two clusters.

6. Conclusions

In this document we described the design and specification of the analysis systems to be integrated with the RAMSES platform. Specifically, after providing a description of the targets malware threats and a review of the analysis tools proposed in the state of the art, we provided an overview of the requirements that we defined in the design of our system. Then, we presented two systems: Prometheus and BitIodine. Prometheus aims at analyzing banking trojans by looking at the artifacts that such malware leaves in memory while performing malicious activity. BitIodine is framework that, parsing the blockchain, aims at clustering addresses that are likely to belong to a same user or group of users, classifying such users and finally visualizing complex information extracted from the Bitcoin network. For both systems, we provided a description of the current implementation, showing the submodules that compose such systems, and defining the inputs that each tool requires, and the outputs that they produce. This information is used to design an interface that will allow our tools to communicate with the RAMSES platform.

References

[1] *D2.2 Report on relevant scenarios*. Mara Mignone *et al*. RAMSES

[2] *D2.3 Report on user requirements for the design of LEAs Tools*. Mara Mignone *et al.* RAMSES