“IDENTIFIYING AND COMBATING CYBER-THREATS IN THE FIELD OF ONLINE BANKING”

Jordi Aguilà Vilà

Computer Architecture

Universitat Politècnica de Catalunya, Barcelona

December 17, 2015

Dissertation framed in the PhD program of the Department of Computer Architecture (DAC)

Universitat Politècnica de Catalunya (UPC)

Director: Manel Medina Llinàs

Contenido

Summary

This thesis has been carried out in the industrial environment external to the University, as an industrial PhD. The results of this PhD have been tested, validated, and implemented in the production environment of Caixabank and have been used as models for others who have followed the same ideas.

The most burning threats against banks throughout the Internet environment are based on software tools developed by criminal groups, applications running on web environment either on the computer of the victim (Malware) or on their mobile device itself through downloading rogue applications (fake app's with Malware APP).

Method of the thesis has been used is an approximation of qualitative exploratory research on the problem, the answer to this problem and the use of preventive methods to this problem like used authentication systems.

This method is based on samples, events, surveys, laboratory tests, experiments, proof of concept; ultimately actual data that has been able to deduce the thesis proposal, using both laboratory research and grounded theory methods of data pilot experiments conducted in real environments.

I've been researching the various aspects related to e-crime following a line of research focusing on intrinsically related topics:

- ✓ The methods, means and systems of attack: Malware, Malware families of banker Trojans, Malware cases of use, Zeus as case of use.
- ✓ The fixed platforms, mobile applications and as a means for malware attacks.
- ✓ forensic methods to analyze the malware and infrastructure attacks.
- ✓ Continuous improvement of methods of authentication of customers and users as a first line of defense anti- malware.
- ✓ Using biometrics as innovative factor authentication.

The line investigating Malware and attack systems intrinsically is closed related to authentication methods and systems to infect customer (executables, APP's, etc.), because the main purpose of malware is precisely steal data entered in the "logon "authentication system, to operate and thus, fraudulently, steal money from online banking customers.

Experiments in the Malware allowed establishing a new method of decryption establishing guidelines to combat its effects describing his fraudulent scheme and

operation infection.
I propose a general methodology to break the encryption communications malware (keystream), extracting the system used to encrypt such communications and a general approach of the Keystream technique.
We show that this methodology can be used to respond to the threat of Zeus and finally provide lessons learned highlighting some general principles of Malware (in general) and in particular proposing Zeus Cronus, an IDS that specifically seeks the Zeus malware, testing it experimentally in a network production and providing an effective skills to combat the Malware are discussed.
The thesis is a research interrelated progressive evolution between malware infection systems and authentication methods, reflected in the research work cumulatively, showing an evolution of research output and looking for a progressive improvement of methods authentication and recommendations for prevention and preventing infections, a review of the main app stores for mobile financial services and a proposal to these stores
The most common methods eIDAMS (authentication methods and electronic identification) implemented in Europe and its robustness are analyzed. An analysis of adequacy is presented in terms of efficiency, usability, costs, types of operations and segments including possibilities of use as authentication method with biometrics as innovation.

Resumen

Este trabajo de tesis se ha realizado en el entorno industrial externo a la Universidad como un PhD industrial Los resultados de este PhD han sido testeados, validados, e implementados en el entorno de producción de Caixabank y han sido utilizados como modelos por otras que han seguido las mismas ideas.

Las amenazas más candentes contra los bancos en todo el entorno Internet, se basan en herramientas software desarrolladas por los grupos delincuentes, aplicaciones que se ejecutan tanto en entornos web ya sea en el propio ordenador de la víctima (Malware) o en sus dispositivos móviles mediante la descarga de falsas aplicaciones (APP falsa con Malware).

Como método se ha utilizado una aproximación de investigación exploratoria cualitativa sobre el problema, la respuesta a este problema y el uso de métodos preventivos a este problema a través de la autenticación.

Este método se ha basado en muestras, hechos, encuestas, pruebas de laboratorio, experimentos, pruebas de concepto; en definitiva datos reales de los que se ha podido deducir la tesis propuesta, utilizando tanto investigación de laboratorio como métodos de teoría fundamentada en datos de experimentos pilotos realizados en entornos reales.

He estado investigando los diversos aspectos relacionados con e-crime siguiendo una línea de investigación focalizada en temas intrínsecamente relacionadas:

- ✓ Los métodos, medios y sistemas de ataque: Malware, familias de Malware de troyanos bancarios, casos de usos de Malware, Zeus como caso de uso.
- ✓ Las plataformas fijas, los móviles y sus aplicaciones como medio para realizar los ataques de Malware.
- ✓ Métodos forenses para analizar el Malware y su infraestructura de ataque.
- ✓ Mejora continuada de los métodos de autenticación de los clientes y usuarios como primera barrera de defensa anti- malware.
- ✓ Uso de la biometría como factor de autenticación innovador.

La línea investiga el Malware y sus sistemas de ataque intrínsecamente relacionada con los métodos de autenticación y los sistemas para infectar al cliente (ejecutables, APP’s, etc.) porque el objetivo principal del malware es robar precisamente los datos

que se introducen en el “logon” del sistema de autenticación para operar de forma fraudulenta y sustraer así el dinero de los clientes de banca electrónica.

Los experimentos realizados en el Malware permitieron establecer un método novedoso de descifrado que estableció pautas para combatir sus efectos fraudulentos describiendo su esquema de infección y funcionamiento

Propongo una metodología general para romper el cifrado de comunicaciones del malware (keystream) extrayendo el sistema utilizado para cifrar dichas comunicaciones y una generalización de la técnica de Keystream.
Se demuestra que esta metodología puede usarse para responder a la amenaza de Zeus y finalmente proveemos lecciones aprendidas resaltando algunos principios generales del Malware (en general) y Zeus en particular proponiendo Cronus, un IDS que persigue específicamente el Malware Zeus, probándolo experimentalmente en una red de producción y se discuten sus habilidades y efectividad.
En la tesis hay una evolución investigativa progresiva interrelacionada entre el Malware, sistemas de infección y los métodos de autenticación, que se refleja en los trabajos de investigación de manera acumulativa, mostrando una evolución del output de investigación y buscando una mejora progresiva de los métodos de autenticación y de la prevención y recomendaciones para evitar las infecciones, una revisión de las principales tiendas de Apps para servicios financieros para móviles y una propuesta para estas tiendas
Se analizan los métodos más comunes eIDAMS (Métodos de Autenticación e Identificación electrónica) implementados en Europa y su robustez y presentamos un análisis de adecuación en función de eficiencia, usabilidad, costes, tipos de operación y segmentos incluyendo un análisis de posibilidades con métodos biométricos como innovación.

Resum

Aquest treball de tesis s'ha realitzat en l'entorn industrial extern a la Universitat com un PhD industrial en l'entorn financer de Caixabank. Els resultats d'aquests PhD han sigut testejats, validats e implementats en l'entorn de producció d'aquesta institució financera i han sigut utilitzats per altres que han seguit les mateixes idees.

Les amenaces més candents contra els bancs a tot l'entorn Internet, es basen en eines programari desenvolupades pels grups delinqüents, aplicacions que s'executen tant en entorns web ja sigui en el propi ordinador de la víctima (Malware) o en els seus dispositius mòbils mitjançant la descàrrega de falses aplicacions (APP falsa amb Malware).

Com a mètode s'ha utilitzat una aproximació d'investigació exploratòria qualitativa sobre el problema, la resposta a aquest problema i l'ús de mètodes preventius a aquest problema a través de l'autenticació.

Aquest mètode s'ha basat en mostres, fets, enquestes, proves de laboratori, experiments, proves de concepte; en definitiva dades reals dels que s'ha pogut deduir la tesi proposada, utilitzant tant investigació de laboratori com a mètodes de teoria fonamentada en dades d'experiments pilots realitzats en entorns reals.

He estat investigant els diversos aspectes relacionats amb e-crime seguint una línia d'investigació focalitzada en temes intrínsecament relacionades:

- ✓ Els mètodes, mitjans i sistemes d'atac: Malware, famílies de Malware de troians bancaris, casos d'usos de Malware, Zeus com a cas d'ús.
- ✓ Les plataformes fixes, els mòbils i les seves aplicacions com a mitjà per a realitzar els atacs de Malware.
- ✓ Mètodes forenses per analitzar el Malware i la seva infraestructura d'atac.
- ✓ Millora continuada dels mètodes d'autenticació dels clients i usuaris com a primera barrera de defensa anti malware.
- ✓ Ús de la biometria com a factor d'autenticació innovador.

La línia investiga el Malware i els seus sistemes d'atac intrínsecament relacionada amb els mètodes d'autenticació i els sistemes per infectar al client (executables, APP s, etc.) perquè l'objectiu principal del Malware és robar precisament les dades que s'introdueixen en el "logon "del sistema d'autenticació per operar de manera fraudulenta i sostreure així els diners dels clients de banca electrònica.

Els experiments realitzats en el Malware van permetre establir un mètode nou de desxifrat que va establir pautes per combatre els seus efectes fraudulents descrivint el seu esquema d'infecció i funcionament

Proposo una metodologia general per trencar el xifrat de comunicacions del Malware (keystream) extraient el sistema utilitzat per xifrar les comunicacions i una generalització de la tècnica de keystream.

Es demostra que aquesta metodologia pot fer-se servir per respondre a l'amenaça de Zeus i finalment proveïm lliçons apreses ressaltant alguns principis generals del Malware (en general) i Zeus en particular proposant Cronus, un IDS que persegueix específicament el Malware Zeus, provant-ho experimentalment en una xarxa de producció i es discuteixen les seves habilitats i efectivitat.

A la tesi hi ha una evolució investigativa progressiva interrelacionada entre el Malware, sistemes d'infecció i els mètodes d'autenticació, que es reflecteix en els treballs d'investigació de manera acumulativa, mostrant una evolució de l'output de recerca i buscant una millora progressiva dels mètodes de autenticació i de la prevenció i recomanacions per evitar les infeccions, una revisió de les principals botigues d'Apps per a serveis financers per a mòbils i una proposta per a aquestes botigues

S'analitzen els mètodes més comuns eIDAMS (Mètodes d'autenticació i identificació electrònica) implementats a Europa i la seva robustesa i vam presentar una anàlisi d'adequació en funció d'eficiència, usabilitat, costos, tipus d'operació i segments incloent una anàlisi de possibilitats amb mètodes biomètrics com innovació.

INTRODUCTION

Online financial services (online banking) allow substantial cost reduction (over 50%) in banking. With the birth of the Internet, the financial sector soon identified a business opportunity and a chance to reduce operating costs in the use of web pages and portals as a commercial channel. The network of commercial Internet began to expand in 1990, and the financial sector was quick to create the first web pages in that decade.

For two decades, parallel to the development of online banking sector, the criminal world on the Internet was developed (e-crime, by its Anglo-Saxon name). This group of criminals needed no scientific explanation, it was just easier to try stealing money online, than wielding a pistol in a conventional robbery.

The organized crime groups were soon interested in this medium as a low-risk way to make money and quick profits. Today many organized crime groups operating from across the world with the aim of stealing money from customers in the financial sector.

Platforms through customers connect to online banking bank have been developed in parallel with the Internet and communication skills and protocols evolution. Therefore the methods of client authentication when using bank services are crucial to prevent fraud. Due to this fact authentication systems and technical systems and procedures used to commit fraud and steal money (malware, social engineering, etc.) are and will be closely related.

The evolution of the first data transport protocols TCP and IP internet protocols, the hardware platforms, And the software made possible the evolution of personal computers and mobile platforms, and this led to a gradual change on the modalities used by the industry e-crime attackers, which have increased, modified, transformed and evolved their systems of attack as the hardware and software developments opened up new possibilities.

The first attacks were directed to communications networks and systems, the evolution have given attackers to the platform (PC) of the final user –the customer-, who is - by excellence- the weakest link in the chain because of its little technical knowledge of computer security. Attacks to users rely on social engineering routing them to become infected with malware, to fake websites (phishing) or false applications (executables, APP's with malicious content) than with the art of deception download malware to their working platforms fixed or mobile (computers, tablets, “smartphones”, etc.).

The financial and payment media sector is one of the critical infrastructures most affected by fraud through electronic channels and new technologies. The challenges for the defense systems of companies and institutions in the public and private sectors and even governments are colossal.

Attacks against banks throughout the Internet environment threats are based on software tools developed by criminal groups, applications running on web environments, either in the computer of the victim (malware) or by using different social engineering ploys to lure potential victims to a fake website (Phishing), or on their mobile devices by downloading rogue applications (false APP), the Malware in all is the application star, due to its effectiveness and ease camouflage.

What in the seventies was a matter of a few with advanced technological knowledge, became a "comodity", actually it’s nowadays possible the purchase of a kit or specific Phishing Malware for the banking sector (a Trojan), designed to steal credentials banking customers at an affordable price. ($ 40 per day, or $ 450 for a month is available a phishing kit; $ 1,200 for a highly sophisticated Trojan kit, or $ 800 to $ 4000 by Zeus toolkit basic Trojan, or $ 100,000 to $ 6,000 by one with the source code, according to rumors, [31] code).

Objectives Assets are monetary client funds, the reputation of the bank and the confidence that customers have in their bank, and by extension, the identity of the customer which, in order to perpetrate attacks, is necessary to replace in many cases. The reputation of the electronic channel as a means for banks is also crucial. The loss of reputation as a channel could end the online banking business.

The methods of attack have been becoming more sophisticated; from crude initial methods increasingly towards sophisticated attacks. Methods are passing through the boom of the phishing Phishing, the current boom of Malware, the APP's fraudulent, targeted attacks and attacks on cloud systems. Attacks are increasing their diversity and sophistication to levels that require high technical knowledge on the part of its developers. Each attack is determined to overcome in one way or another authentication systems used by banks to their customers to operate.

Phishing is a method of attack using social engineering (The Art of Deception) to steal login credentials to bank customers. The most basic method is to send thousands of emails by methods of spam (sending mass mailings), supplanting the Web identity of the bank by some ruse (use of logos and signs of identity) and false message to induce customers to the bank fake page to hand over their credentials.

After obtaining the key the attacker logon on the bench (like legitimate client) and makes transfers to accounts allies (Mules), collaborating in voluntary or also deceived by the same social engineering. The phishing attack methods are diversified from the initial attacks in which keys are requested by the sending e-mail.to more sophisticated methods such as rock phishing (involves multiple domains), the fast-flux phishing (multiple IP), smsphising (sms), spearphishing (phishing proximity or trust), etc. All of them designed to hinder the "take down" - suspension – of the fraudulent pages.

Hence the more sophisticated authentication methods are, the harder it will be to carry out the deception.

We understand for authentication- authentication method or authentification- the act of establishing or confirming something (or someone) as authentic. The process of attempting to verify the digital identity of the sender of a communication as a request to connect to the legitimate website of the bank, is also authentication.

On the website of the bank trustily authentication is a way to ensure that users are who they say they are and therefore authorized users.

The criminals go where the money is. The use of highly sophisticated Malware represents one of the most harmful threats to this sector. Some of the financial institutions that conduct research play an important role in detection and in implementing countermeasures to protect their customers.

One of the biggest threats comes from malware purposely developed to perform fraud in banking systems. The banking Malware (Trojan) is a code that is installed on the user's computer using standard techniques of computer viruses, for capturing data exchanged by the user in his day operation and his bank, including information of authentication and user operation

The most basic are the "keyloggers" that collect, record, read the user's keystrokes or capture images even displayed on the screen when the user operates the banking website. Malicious code continues specializing. Banker Trojans are customized specifically for banks changing the look of your real website, by superimposing a fake window when the user accesses the online service of your bank. The overlay simulates, in part or in full, to the legitimate website so that the user enters information authentication codes in it, believing that it does in the real.

The methods of attack by malware evolve significantly from simple “keyloggers” and defenses authentication methods consisting of simple things like putting a virtual keyboard. To new malware that could quickly perform many more tasks to achieve its objective of removing credentials and send the attacker, to the actual much more elaborated banking Trojans that allowed steal credentials using concealment methods, modifying the operating system and shipping credentials during operations etc. To end as the middleman methods ("Man in the middle") that currently is a sophisticated attack in which the attacker (through its Malware) acquires the ability to read, insert and modify transactions between the bank and its customer without any of the two sides know that the link between them has been violated, or the "Man in the Browser" which is a type of attack using malware (trojan), after infecting a machine that is capable of modifying pages websites, content or transactions, in an invisible way for both the user and the bank server.

In the financial sector we need to quickly detect, understand its mechanisms, and neutralize its infrastructure to prevent fraud consumer, prevent money laundering and protect customers from Trojans.

Also affected platforms have evolved from networks, workstations and personal computers, to be any mobile phones, smartphones, tablets, iPods, ipads, iphones, etc. currently affected by new generation Trojans.

There are several research groups both in educational institutions and in the private sector. The first publications on security in the e-crime sector talk were about security on computers, networks and applications. Later mainly due to the development of smart phones publications on security in mobile applications and APP's appear.

1.1 Environment conducting the research

This PhD work has been performed in an environment outside the University, like an industrial PhD. The hosting and funding organization has been CaixaBank, one of the leading financial institutions in EU and Spain. The results of this PhD work have been tested, validated and implemented in the productive environment of that financial institution, and others have followed the same ideas.

"La Caixa" suffered the first phishing attack at the beginning of 2005 and as a defense for future attacks, after this first break up, I established a group of incident response. e-LC CSIRT. (E-Lacaixa Computer Security Incident Response Team). It was in this environment in which it became clear the need to develop a parallel field of research that allow us to understand better, systems, schemes, utilities, tools and processes related to crime on the Internet (e-crime). The need to investigate the strengthening of our defense systems arise same time; the client authentication methods to which these attacks are directed.

The first malware attack was a "0 day". A virus that had no reference in the wild arise and attacked the bank and the industry a few months later. We got a sample of this attacker malware at the home of a client, we performed forensic analysis, we realized how it works and we get dismantle their infrastructure. Subsequently delivered to antivirus firms worldwide vaccine was manufactured. The virus was baptized with the name "Anserin".

To effectively counter mesure these threats it became apparent the need for a thorough understanding of the operation of banking malware. To stop it I've created an internal research group in the CSIRT (Computer Security Incident Response Team), focused solely on research of various aspects of the viruses, Trojans, forensics, incident response systems for Malware and investigate the authentication means to counter them.

This group established national and international contacts. Cooperation in universities and other relevant research sought worldwide; These include Carnegie Mellon University, Polytechnic University of Catalonia, Universitat Ramon Llull, Trinity College Dublin, Southern Methodist University, Baylor University, SEI (Software Engineering Institute in Pittsburgh), University of Alabama at Birmingham, SANS Institute, etc.

The same year the CSIRT became a member of FIRST (Forum of Incident Response & Security Teams) and in front of APWG (Anti-Phishing Working Group). At FIRST I spent three years on the Board of Directors and currently I' am a member of the Board of Directors and the Steering Committee of APWG, Inc. and founder and patron APWG.eu.

Subsequently various alliances with other research groups, UDC (University of Dublin), FBD (Barcelona Digital Foundation) and the interests of Research and Development from BDigital and CSIRT converged and new alliances were made, expanding the research group for issues of e-crime.

It is in the context of defending the interests of clients in the financial sector and the entity itself deeply concerned for the safety of banking online and the effect on the reputation of this business, where I started several investigations, directed by me, with the participation of people in my team, in collaboration with renowned research estates, always related to the world "e-crime", in order to figure out the various aspects of the criminal underworld of Internet, related Malware banking and barriers that must overcome these applications: the authentication systems.

1.2 Vision

The aim is not just pure research itself, if not applied research to medium or long term, or even conduct the banking sector to a better understanding about how this criminal underworld works and help this sector to improve, prevent and counteract consequences of the activities of the criminals having focused their activities in this direction.

I've been researching the various aspects related to e-crime following a line of research focusing on intrinsically related topics:

- The methods, means and systems of attack: Malware, Malware families of banker Trojans, Malware as case of use, Zeus as case of use.
- The fixed platforms, the mobile applications used and as a means for malware attacks.
- forensic methods used to analyze the malware and infrastructure attacks.

Continuous improving of authentication methods implemented to ensure customers and users as a first line of defense anti- malware used by banks.

The utilization of biometric system implemented as innovative factor authentication.

We conducted various researches on malware in general, the various families of malware (Trojans) that attack the institutions of the banking sector and the means of defense against this system of attack by malware. Also these and other attacks occur with the involvement of the use of social engineering against authentication methods, which is the reason why the use and development of these methods are vital for security professionals in the banking sector.

Within this framework, my thesis studies the banking malware in general and families of malware (ZEUS) as case of use, and authentication systems in production in major European companies facing this malware and the use of biometrics as an authentication method for online banking as a case of use.

1.3 Motivation: questions the thesis should investigate.

1.3.1 Motivation & reason 1

General financial malware research topic: How the Financial malware work? What platforms and electronic environments attack? How to carry out their attacks? How does the Zeus malware family attacks, How we can detect it, analyze it from the point of forensic view and how we can design tools to neutralize the threat posed to customers of the banking sector? What platforms, electronic environments, applications it attack (phones, computers, "tablets", etc.)?

1.3.2 Motivation & reason 2

Malware affecting banking: What mechanisms, products, systems can be used to neutralize it? We propose "Cronus" IDS specifically fighting Zeus. What we give as recommendations to prevent attacks in fixed, mobile platforms? What role does apps, mobile applications? What recommendations we can provide against Malware.

1.3.3 Motivation & reason 3

Current methods of authentication: What are the methods used by professionals? What are the reasons for use? What segments? what parameters are used? Use reasons: robustness, usability, efficiency, etc.?

1.3.4 Motivation & reason 4

Secure authentication: What we can recommend as more secure authentication methods depending on market segments-customers, electronic platforms (phones, computers, tablets, etc.), to counter attacks by Trojans and other methods? Do we have to use of biometrics as anti-Malware barrier?

1.4 Objectives of the thesis. With the work done in this thesis is pursued:

1.4.1 Objective 1:

Understand the functioning of financial Malware families Malware finance, the Zeus Trojan family as a case of use study and example of other future financial malware families.

1.4.2 Objective 2:

Give a set of recommendations and a tool to become more effective combating malware in both fixed and mobile platforms. Make recommendations to neutralize attacks on mobile platforms and the Apps.

1.4.3 Objective 3:

Analyze authentication systems of banks. Propose alternatives. Study the use of biometric authentication methods.

1.4.4 Objective 4:

Give a series of recommendations to make the system more effective attacks against malware using authentication methods Analysing "eFraud" including different families of malware, including methods of attack and their barriers we built trough the authentication methods.

2. RELATED WORK. “STATE OF THE ART”

There are several forums where different authors conduct research scientific publications in this field.

There are also private associations whose goals include research in the field e-crime;

- APWG (Anti-Phishing Working Group)
- FIRST (Global Forum of Incident Response and Security teams)
- Mobey Forum,
- MAAWG (Message Anti-Abuse Working Group)
- Cloud Security Alliance,
- ICANN (Internet Corporation for Assignment Names and Numbers), etc.

The bibliography reflects the problems of safer use of the Internet since the beginning of the commercial use of financial services such as interaction channel between the client and the bank.

The literature review of the issues presented in the thesis reflects the various vectors of research.

Regarding the research work carried out concerning the problem posed by malware; in recent years financial malware and fraudulent activities have been studied in various aspects. Chandrasekaran et al. [1] propose an approach for detecting phishing attacks using false responses and monitoring the behavior. Birk and Gajek [2] extended a framework that uses "honeytokens" [3] to track phishers. Technology-based strategies have been developed to analyze botnets and disable dynamically. Holz et al. [4] they introduced a methodology to track and observe botnets using "honeypots". Inspired by the work of Holtz, "The Dorothy Framework" [5] was developed with the intention to track and display a botnet automatically. This research framework was customized [6] to investigate botnets oriented to financial sector. "Comand and Controls" (C & C) - - In modern malware routine encrypted to encrypt their traffic to communicate with its control systems are used is necessary to remove the symmetric key communication Binary Malware to succeed in deciphering flow communication. For this purpose, Caballero et al. [7] proposed a tool that takes advantage of the binary code to extract fragments of code encryption / decryption and reused to discover malicious traffic.

Leader et al. [8] [9] came to the same objective monitoring data exchange malware to level I / O interfaces of the sandbox.

Alongside the research efforts in analysis and detection of malware they have conducted research on countermeasures to counter the threat. Blocking IP to DNS has been investigated frequently [10] [11] [12] to prevent communication of those infected with the C & C with different modalities that resolve to a non-accessible or blocked IP computers. On the other hand they have been used in the identification patterns IDS to identify and block malicious traffic from botnets [13] [14]. Omerod et al. [15] investigated kits botnets, reputational botnets to discourage or pursue to the end user of the stolen credentials. Ford and Gordon [16] focused their research on methods of income attacking malicious code to harm the economic model of criminals. Focusing on the case of use Zeus, Binsalleeh et al. published a comprehensive analysis of the 1.2.4.2 version of the tool [16]. They did reverse engineering to discover its internal characteristics and to better understand their behavior to stop injecting false information to the C & C botnet and consequently damage the reputation of malware. In its publication they indicate that the communication between the infected host and Zeus C & C resides in a vulnerable implementation of RC4 encryption that can be exploited by an attack flow reuse keys.

Our research in Malware expands their research into new techniques for deciphering Zeus findings. In "Taming Zeus by leveraging Its Own internals crypto" Our research is based on analysis of the communications flow behavior. Additionally, our work focuses on a new version of Zeus that has a higher level of complexity with respect to the publication specified. In Titans' revenge: Detecting Its Own Zeus via flaws "investigated the vulnerability given by authors [17] By focusing on the analysis of the host. This extensive work previous research results of Zeus, researching new techniques to detect and decode communications Zeus. Also Zeus version is newer than the previously published studies. Historically criminals go where the money is and the impact of malware attacks in the financial sector and its mode of operation can be seen clearly citing as examples the two high impact attacks carried out in 2012, the "High Roller incident" [18] that he got fraud 60 million Euros, and the "Eurograbber" which were stolen 36 million Euros in 30,000 banking customers [19]. In both attacks hackers managed to take control and authentication keys subtract, mocking systems

two-factor authentication and perform fraud. As a result, authentication systems continue magnetically attracting attention from research groups and professionals.

Concerning the Malware and its relationship with the authentication methods used in banks in the European financial business sector, it considers that, as stated in the above examples, authentication systems play a key role in the successful execution of a malware attack and the weaker is an authentication system many more possibilities with the attackers to get the money, as well, considering that cyber criminals are increasingly organized and sophisticated in new technology groups must continually identify Emerging challenges associated with technological developments.

In [20] authors discuss overall challenges and relevant aspects identified in e-banking. Other authors [21] [22] have presented an extensive study on e-banking security and introduce a formal definition of threats in e-banking and security models. In [23], the author presents a study on security threats through the proposed use of biometrics for strong authentication in e-banking scenarios. Taxonomy of attacks on authentication systems in e-banking is introduced by the authors in [24], further authentication solution based on "challenge / response" is proposed. An assessment methods authentication for e-banking by the authors proposed in [25], however the presented study focuses only on the PC environment. Authors in [26] analyzed the eIDAS systems used in major banks in English-speaking countries; the investigation is limited to data collection by observation (published data). Contrary to our approach, which analyzes these same factors eIDAS considering current e-banking with information provided by professional systems leading European banks? Finally our research provides advice on eIDAS and a set of recommendations that take into account the perspective of security professionals.

3. RESEARCH WORK CARRIED OUT & METHODOLOGY

3.1 Organization and structure of the research

The main results of this thesis are presented in chapters. As supplements to digest for the thesis we have presented research results at conferences (proceedings and presentations) and papers published by diverse editors.

Method has been used is an approximation of qualitative exploratory research on the problem, the answer to this problem and the use of preventive methods to this problem through authentication. We achieved conclusions and recommendations.

This research method needs no prior assumptions and has been used as a method in other publications. It is based on samples, events, surveys, laboratory testing, proof of concept, in short real data that has been able to deduce the thesis proposal, using both laboratory research methods based on information theory; survey data, data from experiments in controlled environments and data pilot experiments conducted in real environments.

The thesis is the response to the question and the objectives we have settled at the beginning of the research. Through its chapters responds to the thesis problem and questions and gives tools, recommendations and solutions to it. At the same time gives a progressive evolution trough the research to the problem.

The publications done during research are an evolution of the research with related topics. Malware investigating the line and attack systems intrinsically related to authentication methods and systems to infect customer (executables, APP's, etc.) because the main purpose of malware is precisely steal data entered in the "logon "authentication system to operate and thus fraudulently steal money from online banking customers.

That is why the tasks of the research are to investigate the malware in general, the most famous family of malware Zeus as use case, the APP's phones that include Malware and research tasks of authentication methods and the incorporation of biometrics in such methods as the case of use.

In the line chosen there is an interrelated research gradual evolution from malware infected systems and authentication methods-methods anti Malware defense and prevention - which is reflected in the thesis work and its publication publications cumulatively, showing an evolution of research output and looking for a progressive improvement in the results of research on malware and improved authentication methods and prevention and recommendations to avoid infection.

The thesis is organized in chapters focusing the problem of the hypothesis. Chapter one explains about malware analysis and solutions, chapter two explains about authentication methods and use as Malware solution, chapter three is about results and conclusions. Contributions to the research on the diverse aspects are largely commented trough the chapters.

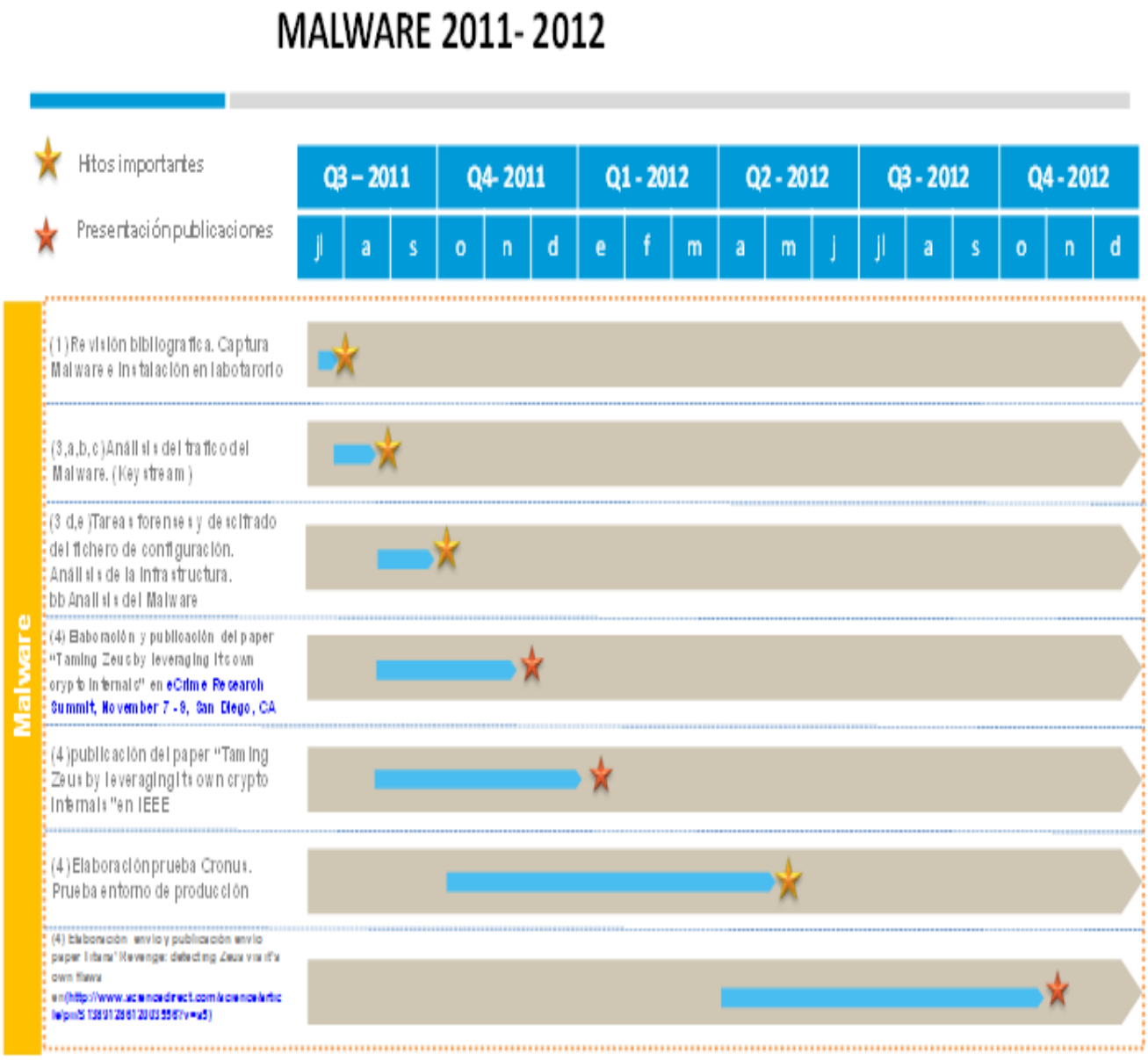
3.2 Malware.

The reason why malware is chosen generically is because together with phishing are most concerning threats in the financial sector and more truble us as security professionals in the banking sector, with one notable difference. Phishing in all "visible" and the response to mitigate risk is easy to deduce, imagine and apply. By contrast in malware attack system is completely hidden in nature besides obfuscated on purpose, we must make a considerable effort to understand the intricacies of its operation before displaying even the first steps to mitigate them. It is here that play a crucial role authentication methods, investigated in the work. As more sophisticated they are more effective they are, but this fact is working against other aspects like usability.

That is why in our first research on Malware was selected Zeus as a case of use, whose attack system, widespread in other banking Trojans and network with more or less sophistication in its different versions, was focused on getting the credentials of our e-

banking customers, alongside the many systems and financial institutions in the world that have similar systems to ours consisting of a first level authentication ID and pin six positions for communicating with the bank and a second level with random coordinates card 60 positions.

Why Zeus as use case? The reason is simple Zeus or Zbot, born for the first time in 2006 at the hands of Russian cybercriminals is the world's most famous banking malware due to its sophisticated modus operandi. In July 2009, at its peak, a safety report Damballa [29] placed him number one ranking as the main threat among all Botnets, infecting 3.6 million computers in the US alone.



The work carried out so far has materialized in different approaches described in the first thesis chapter. Its evolution has conditioned our authentication systems in the wide range of the industry, fostering the change and evolution of the different authentication systems to face it and achieving in our case to add an OTP system, based in Challenge-response as final query to complete a transfer operation.

During the thesis research, six main publications have been achieved and can be found in table that we include as a demonstration of the work achieved and as a dissemination method of our results through the scientific community worldwide working in similar topics. In addition, we add conference presentations specified in Tables 1 and 2.

3.3 Methodology used to address the problem of Malware.

3.3.1 Establishes the state of the art through the literature review.

3.3.2 To understand how the infrastructures that support the Malware work we realized a previous work where we will study various botnets functioning

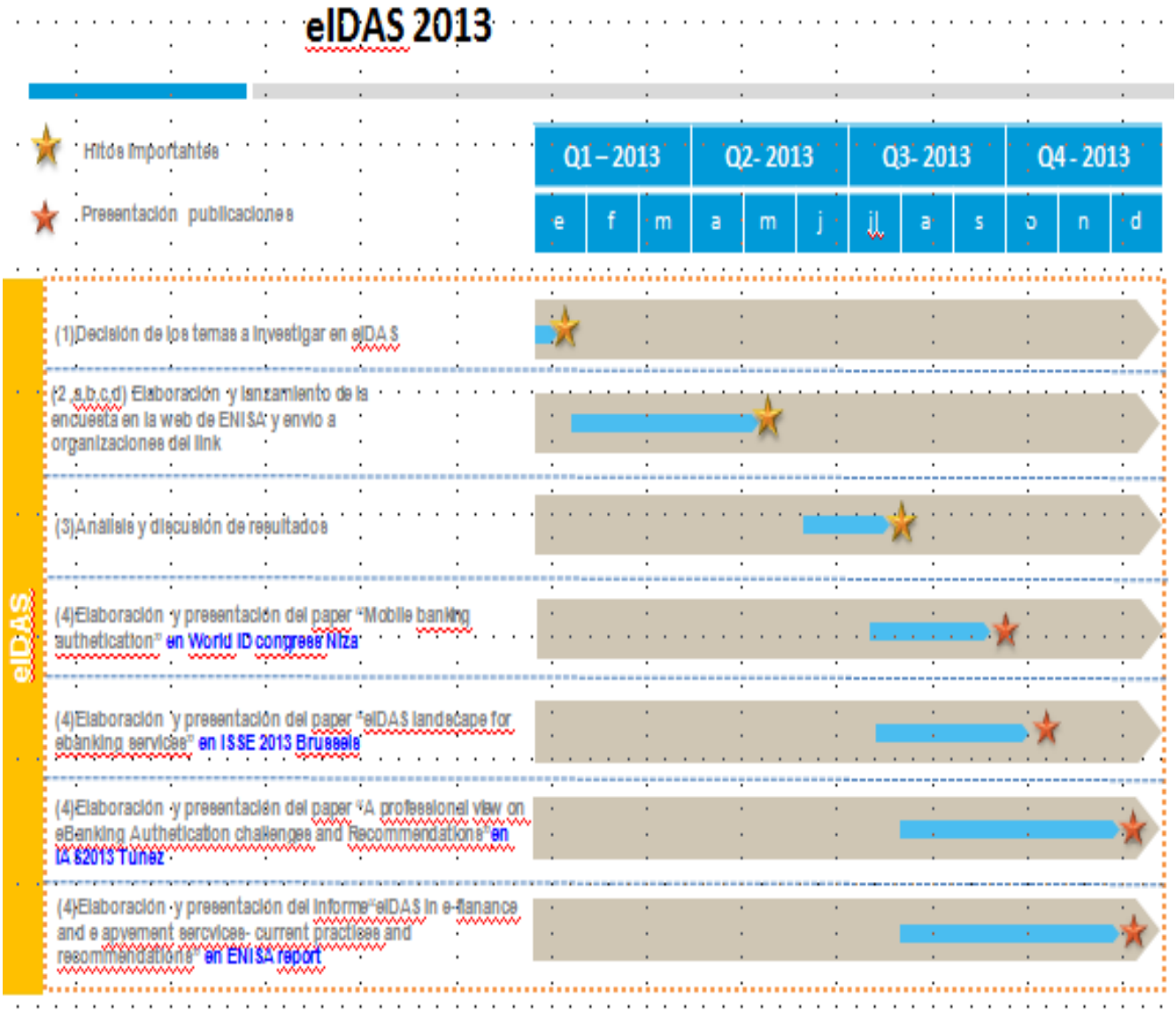
3.3.3 To study the malware we will do the following experiment with the following:

Samples

3.3.3.1 Capture of samples in customers' computers infected by Malware banking.

3.3.3.2 I have defined privacy policy and technical methodology in the capture of samples, validated the process of collecting them. A team of e-LC CSIRT is responsible for identifying customer contaminated equipment and extracting samples under my supervision.

3.3.3.3 Install the samples in our laboratory machines that I get infected with malware (of the ZEUS family). If within the time frame set for the completion of the thesis new families of malware arised, and it is possible to get samples of malware that appear on the Internet, the experiment will be extended to the new samples and will follow the same pattern, although it may we should modify some aspect of flow analysis as forensic breakthrough phase.



3.3.3.4 I led the definition of the architecture used to analyze the malware systems and sources have identified where to get samples of malware and managed collaborative arrangements with providers of the same. Also under discussion with the team we have chosen the software platforms used for the experiment.

3.3.3.4.1 Analyze malware traffic generated Keystream techniques. This analysis has made the team, under my supervision.

2.3.3.4.2 Analyze the malware download files on the computer victim. This analysis has made the team, under my supervision.

3.3.3.4.3 Forensic analysis and drawing conclusions from the analysis of the malware. Forensic testing has been performed by the task force, under my supervision. The conclusions are agreed in team meetings led by me.

i. I’ve been identifying methods to counteract the consequences of the attacks, through contacts with professionals and researchers in international forums, and

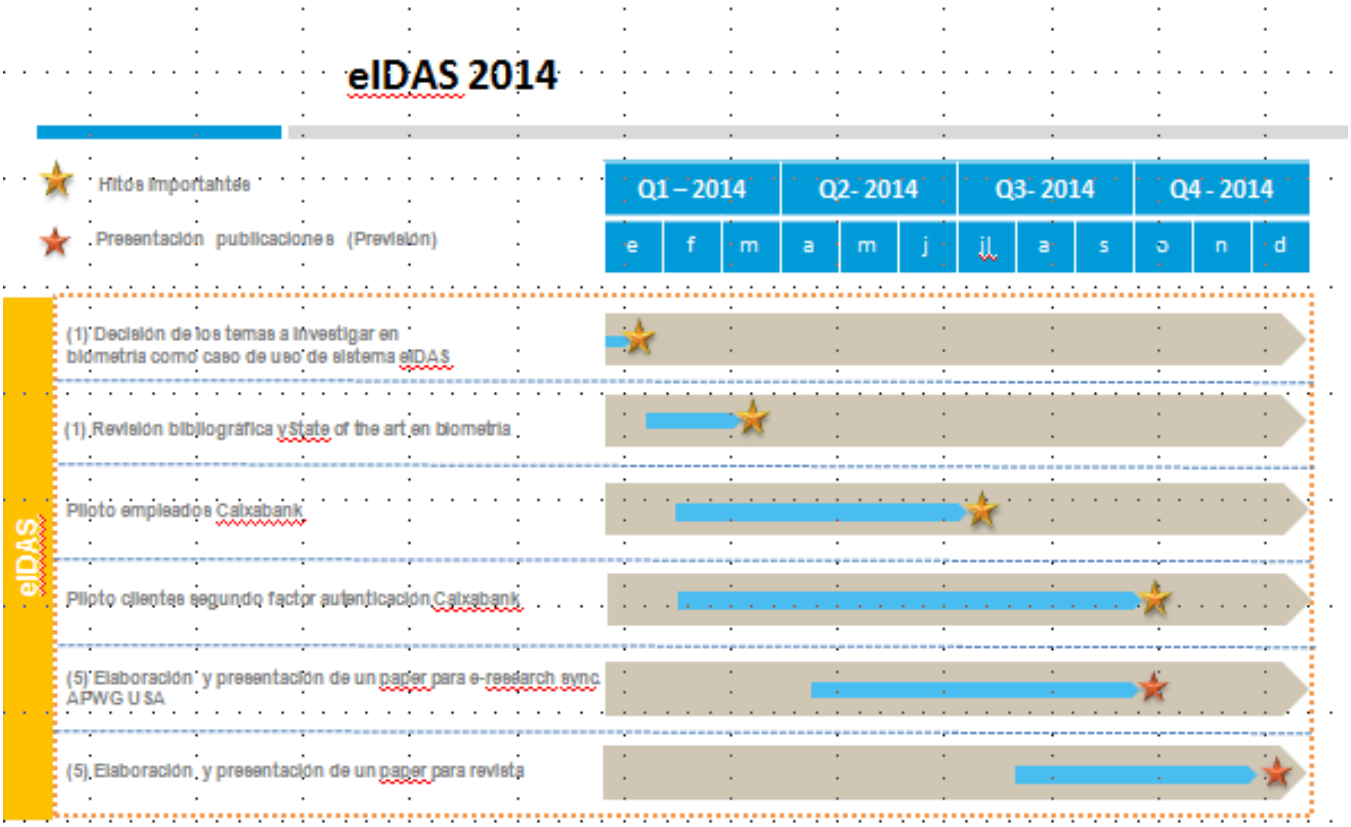
ii. I’ve been making proposals for improving these methods, which constitute innovative contributions in cyber security in the financial sector.

New mechanisms and new methodology to neutralize the malware is proposed.

We implement proofs of concept. We recommend on how to combat malware through various systems and best practices derived from lessons learned from the analysis and publication of results of research work done.

3.4 Methodology to address the issue of authentication systems

3.4.1 Establish the state of the art through the literature review.



3.4.2 To address the research on authentication systems during 2013 conducted a survey directed to professionals in the banking sector in Europe; CISO's, CIO's, IT Managers security. This survey was launched through various forums with which I have contacts. APWG, FI-ISAC, ENISA, FIRST, etc.

3.4.2.1 Development of the contact list. This task has been undertaken by my thanks to the contacts made in many years of work in international organizations in the security sector and positions held in the management bodies of FIRST, ICANN and APWG.

3.4.2.2 "triage" of the topics of the survey. This task is done by the team with my direct participation and supervised by me.

3.4.2.3 Development of survey questions. This task is done by the team with my direct participation and supervised by me.

3.4.2.4 Release of the survey contacts. This task has been undertaken by ENISA and supervised by me. From the results obtained in the survey will do various jobs to consider which are the most commonly used authentication methods in European banking, reasons for use, segments which are implemented and to recommend its use.

4. CHAPTER I: Malware

4.1. The Zeus malware

The Zeus malware (also known as Zbot) first appeared in 2006 when a security firm released a full reverse engineering analysis [27] about an unknown trojan named PRG. Since then, it has been modified and customized to suite specific needs and released in different variants, each one offering innovative features to steal sensitive information. Zeus was originally created, distributed, and maintained by Russian cybercrime gangs [28]. Historically, the russian cyber underground scene is mainly of criminal intent, as the ultimate goal would be to maximize the amount of money the participants could make. On top of that, the general hacking environment in Russia can be mainly characterized as financially driven. Controversially, legal persecution of cyber crime in Russia is not a priority. Strategically, Russian hackers usually avoid to target regular Russian citizens in order to gain a shared sense of toleration and even admiration. In addition, the strategic choice to prefer targets outside the Russian Federation complicates the cross-border cooperation that is needed to investigate the cybercrime related frauds. Indeed, investigating crimes against foreign interest does not represent a priority for law enforcement officers in Russia [28].

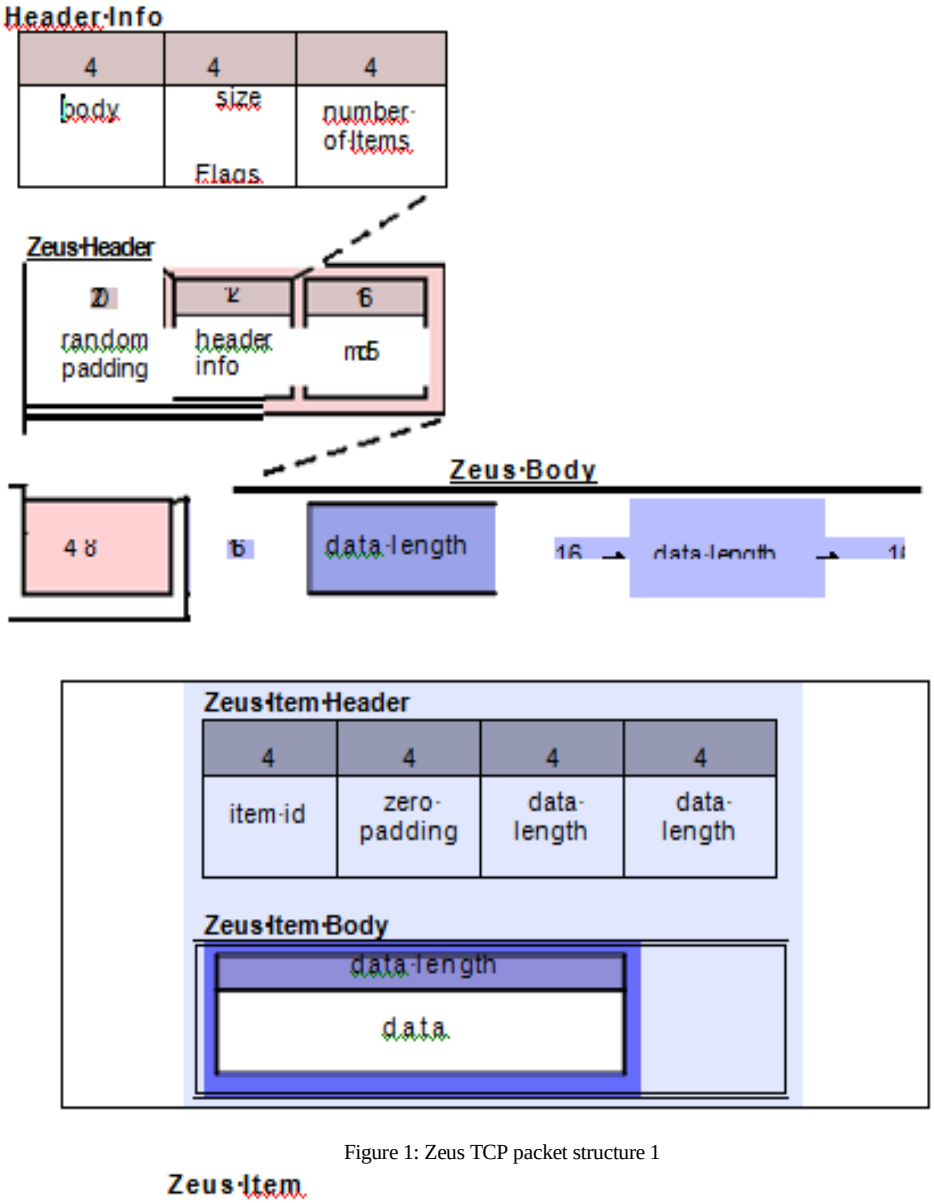


Figure 1: Zeus TCP packet structure 1

4.1.1. The crimeware toolkit

The Zeus crimeware toolkit used to be sold privately for a price that ranged between $800 and $4,000. However, its mar-ket value has been drastically reconsidered since the leakage of its source code on March 2011. Interestingly, on February 2011, a forum post was published on a notorious underground forum where a seller offered the full version of the Zeus source code (toolkit included) for a “high price”. Speculations suggest that the price asked by the seller was floating around $100,000 [31]. When the source code was leaked, it generated a spiky increase of new Zeus malware variants, offering to the entire Internet community a framework to easily set up a robust financial botnet for free. The builder offers the capability to create customized malware executables and the botnet configuration files needed for the correct botnet operation. In this way, users could create their own malware which targets the financial institutions that have been inserted in the malware building process. Additionally, the control panel offers the ability to easily ad-minister the botnet trough a user-friendly PHP page that allows the botnet owners to quickly retrieve a comprehensive status of their bots, and to be able to download the stolen information. Moreover, the installer comes with a bilingual manual (Russian and English) which explain the fundamental steps required to install the Zeus suite.

The Zeus malware is software designed to make a profit also by its development. As a matter of fact, its full customizability offers to cyber-criminals the opportunity to develop new modules and sell them to the market place. For instance, the form grabber module for Mozilla Firefox and the back connect module (the latter offering the botnet owners to have a direct access to the console of the infected computer), were sold for around $2,000. People interested in particular features can post their advertisement on such underground forums, and offer money to those who could develop the requested customized software.

4.1.2. Zeus communication protocol

Like most banking trojans, the Zeus’s goal is to steal sensitive information that could lead the attacker to carry out a financial fraud against the victim. The Zeus ecosystem is usually composed of three different entities: the bot, e.g. the machine that has been infected, the Command and Center — here in after C&C, or *dropzone* — i.e. the main server where the control panel is hosted and where the bots send the stolen information, and the configuration server: the server where the configuration file is hosted, ready to

be downloaded by the bots. The C&C and the configuration server usually overlap their role, offering to botnet owners the comfort of administering only one server.

Once the victim's computer is infected, the malware hooks every API call in order to grab sensitive information before it is sent through the network. In this way, the malware is able to steal HTTPS sessions before they are encrypted, and also to send them to the C&C. Stolen information mainly resides in HTML input data forms, POP, FTP accounts credentials, X509 certificates stored in the browser, and cookies saved in the sys-tem. Occasionally, the botnet owner can also request a screen snapshot to the owned bots, forcing them to send a screenshot of what they are currently looking at.

Stolen data is regularly sent to the botnet's *dropzone* through two different communication channels. The first one, referred in the Zeus configuration file as *log*, consists in a small keep-alive message containing all the main status information about the zombie i.e. *botID*, *botnetID*, IP-address, bot OS, etc. Notably, this packet is sent to the *dropzone* every two minutes by default, and consists in the most frequent communication type between the zombie and its C&C. Thus, its periodic emission could be leveraged by an anomaly based IDS to identify an infected computer inside a network. The second communication flow used by Zeus, referred as *report*, occurs less frequently than the *log* one i.e. by default every ten minutes. As well as the zombie status information included in the *log* packet, it contains also all the data that has been stolen in the system. Hence, this message is usually bigger than the previous one, and packet fragmentation according to the network MTU size is often required by the OS for sending the entire TCP segment.

The TCP packet structure of the Zeus *report* communication is shown in Figure 1. The packet begins with a 48 bytes long *header*, followed by *n* bytes containing the *body*, which is divided into several *items* containing all the stolen information categorized by prefixed labels. The *header* consists of 20 bytes of random padding, followed by 12 bytes used for the *header info* and 16 used to store the MD5 hash of the Zeus *body*. The *header info* is composed of three slots of four bytes each, respectively containing the Zeus *body size*, the Zeus *item flags* and the number of Zeus *items* contained in the *body*. Zeus *item flags* are used to indicate whether the Zeus *body* is compressed and, if so, how to manage contained items. The MD5 hash is used by the control panel as integrity verification: once the C&C receives the message, it firstly calculates the MD5 of the *body*, and then compares it with the one stored in the packet. If the two hashes match, the packet is processed by the C&C; otherwise the packet is dropped.

Each Zeus *item* is composed by a 16 bytes long *item header* and an *item body* with a variable length. The *item header* is divided into four 4 bytes long slots containing the *itemID*, a zero padding string, and the length of the *item body*. Note that the last two slots represent the same information — a clear rational for this is currently missing. The *item header* is used to identify the information by its type and to anticipate the length of its body. In this way, the Zeus control panel knows how to dissect the received packet, and it is able to store each *item* in the correct SQL field. Table 1 shows the main Zeus *item IDs* used to identify the stolen information. The values of *items 10002* and 10003 are *constant*, and are fixed at the time of the malware creation[1]. We refer to these items as *trojan items*. As for the other *items*, they are variable and strictly depend on the infected system — we refer to them as *environmental items*. It is important to note that all the stolen cookies are stored into the *item id* `10009`, which can reach several Mbytes of length. Indeed, the more cookies the system has, the bigger the entire packet will be.

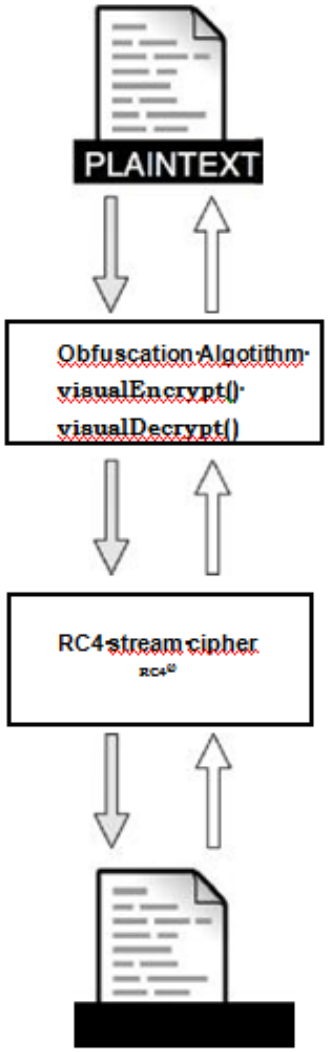


Figure 2: Encryption/Decryption flow of Zeus 2.0.x

[1] While *item* **`10002`** is specified by the bot master, *item* **`10003`**, which defines the bot version, is hard coded within the Zeus Builer toolkit.

ItemID	Value
10001	SBCID BOT ID
10002	SBCID BOTNET
10003	SBCID BOT VERSION
10005	SBCID NET LATENCY
10006	SBCID TCPPORT S1
10007	SBCID PATH SOURCE
10008	SBCID PATH DEST
10009	SBCID TIME SYSTEM
10010	SBCID TIME TICK
10011	SBCID TIME LOCALBIAS
10012	SBCID OS INFO
10013	SBCID LANGUAGE ID
10014	SBCID PROCESS NAME
10015	SBCID PROCESS USER
10016	SBCID IPV4 ADDRESSES
10017	SBCID IPV6 ADDRESSES
10018	SBCID BOTLOG TYPE
10019	SBCID BOTLOG

4.1.3. Zeus crypto routines

Since its first version, the Zeus trojan encrypts the network communication with its C&C in order to avoid identification by pattern-recognition algorithms used by the IDS. The Zeus 1.x malware version used to encrypt its data using the RC4 stream cipher [17], which offers a fast and lightweight encryption in terms of CPU consumption. Indeed, since one XOR operation over a PC *word* can take from four to eight CPU clocks to be completed, a 2GHz CPU could successfully execute from 250M to 500M XOR/s. Besides performance, the RC4 algorithm is also very easy to implement, considering that it can be written from scratch in just few lines. The RC4 *seed* the security of the encryption algorithm relies upon is set by the botnet owner during the malware sample building phase. Once the malware is executed, the seed (also *key* in the following) is stored in a pre-defined and fixed memory space, ready to be used every time the bot needs to download a new configuration file, or needs to send the stolen information. Notably, the RC4 initialization vector is reused at every invocation, meaning that every out-going packet will be encrypted starting from the beginning of the keystream. The research results provided in this paper were a direct result of leveraging this

implementation weakness, al-lowing us to reuse the reconstructed parts of the keystream to decipher the configuration file.

For previous Zeus versions, researchers used to have several ways to automatically find out the key and to decrypt the traffic between the bot and its C&C. As long as the encryption key was stored in a fixed address space, the researchers could automatically dump all the volatile memory of the infected machine and extract the data contained inside a known range space. When the second version of the Zeus malware came out, it introduced a new layer of encryption and a new way to dynamically store its key in memory. Because of this, all the automated frame-works developed around Zeus 1.x needed to be completely re-viewed.

The version analyzed in this paper, Zeus 2.0.8.9, further introduced a new layer of obfuscation on its data communication process between the infected machine and the C&C. Developers who added this additional obfuscation layer probably wanted to avoid IDSs from detecting known patterns previously learnt by Zeus 1.x malware analysis. The sequence diagram in Figure 2 summarizes the encryption and obfuscation routines used by the analyzed Zeus malware. Thus, after decrypting certain RC4 Zeus network traffic, an additional decryption function needs to be invoked to break the entire cipher ring — enabling recovery of the corresponding plain text. Notably, our research demonstrates that this additional encryption layer does not really hinder cryptanalysis. As outlined in further sections, this finding allowed us to identify certain malicious traffic pat-terns without necessarily breaking the obfuscation layer.

At the time of writing, there are several Zeus variants in the wild, due to the source code leakage. Some of them introduce the same domain flux technique used by Confiker and Torpig to evade DNS *sink-holing*, and P2P communications as main communication channel between the bots and their C&C. However, apart for a specific variant that uses AES instead of RC4[34], the encryption mechanism has not been drastically altered from the previous version, preserving the applicability of our findings also to the latest Zeus version.

4.2 Zeus Control Panel

The HTTP control panel is in charge of decrypting incoming communications, sent by the botnet's zombies to their C&C, and to encrypt the relative answers. Its code is written in PHP, and includes three pages in its root directory: `cp.php`, `gate.php` and `install.php`. The leaked Zeus 2.0.8.9 source code was analyzed, taking advantage of

the fact that all the code is well commented at every function point and, although comments are written in Russian/Cyrillic, it was possible to obtain a reasonable translation using the Google Translate web service [33].

The install.php file is an executable page that automatically configures the server environment with the malware requirements. The installation process takes a few minutes: once the botmaster has inserted the environment information into the install page (such as DB server credentials and user admin pass-word) the script takes care of filling in the database with the SQL schema needed for the botnet control panel. Besides the installation page, cp.php is the main page where the botmaster logs into for controlling the botnet. This web page mainly represents the status of the botnet by querying the mySQL database, and offers the capability to send custom commands to the owned bots. The PHP page handling the incoming HTTP POST messages is gate.php, where our attention was particularly focused on.

The code inside this page is in charge of decrypting the ciphered TCP flow incoming from the botnet zombies, and to dissect the payload into different variables, as previously outlined. Further-more, the code takes care of filling in the database with the retrieved information in clear text. The decryption functions used in this web page are stored in an external file, global.php, which resides in the system directory. The obfuscation and the decryption routines detailed later on were analyzed by closely studying these two files.

4.3. Malware analysis methodology

In the current section, we formally define the proposed key extraction and malicious traffic detection methods. Let $v1$ and $v2$ be infected computers running in a controlled environment and belonging to a botnet $bt1$. Let $envv1,i$ and $envv2,i$ be Zeus *environmental items* of $v1$ and $v2$ respectively — remind that those values are known. Let also $\omega bt1$ be the Zeus *trojan item*, with value and size unknown to us. The plain text of a Zeus *report* issued by $v1$ is composed of:

$Rv1 = [header, envv1,0, \omega bt1, envv1,1, ..., envv1,l]$

where the *header* is unknown and depends on the *body*, which is composed of $envv1,i$ and $\omega bt1$. Analogously, the plain text $Rv2$ is composed of:

$Rv2 = [header, envv2,0, \omega bt1, envv2,1, ..., envv2,l]$.

Note that the *item* $\omega bt1$ is common to all bots in the botnet $bt1$. Moreover, reports issued by bots belonging to the same botnet follow the same structure. Each of these

reports is then encrypted in order to obtain packets *C* and *D*, respectively issued by *v*1 and *v*2.

By merging this information with the knowledge acquired during the Zeus Toolkit code analysis, we can identify certain parts of the plain text message, i.e. $env_{v1,i}$ and $env_{v2,i}$, that will be encrypted and sent to the C&C. As discussed before, the Zeus malware does not update the RC4 initialization vector, exposing its communications to key reuse attacks. This weakness permitted us to develop a *chosen-plaintext attack*, based on the known *environmental items*, against the encrypted stream that flows between the bot and its C&C. In particular, we were able to retrieve the keystream —from the cipher text—, and to reuse it to detect infected network traffic. To this goal, we focused on the biggest encrypted packet that a Zeus infected computer periodically sends to its C&C, which includes a full report of the stolen information.

Algorithm 1 Obfuscation algorithm

```
1: function VISUALENCRYPT(data)          ▷ The plain text data
2:     for i ← 1, |data| do
3:         data_i ← data_i ⊕ data_{i−1}
4:         i ← i++
5:     end for
6:     return data                       ▷ The obfuscated data
7: end function
```

Algorithm 2 De-Obfuscation algorithm

```
1: function VISUALDECRYPT(data)  ▷ The RC4 deciphered data
2:     for i ← |data|, 1 do
3:         data_i ← data_i ⊕ data_{i−1}
4:         i ← i++
5:     end for
6:     return data                       ▷ The plain text data
7: end function
```

4.3.1. Cryptanalysis

The RC4 is a stream cipher that uses a bit-wise exclusive-or (XOR) operation between the plain text (*P*) and the keystream (*K*) generated by a pseudo-random number generation algorithm (PRNG). Note that the XOR is subject to the cancellation propriety: $P \oplus K = C$; $P \oplus C = K$.

The cipher text C is contained inside an HTTP POST request made by the infected computer to its drop zone, therefore the entire data payload needs to be extracted from

the network flow before being correctly decrypted. However, the new version of Zeus introduces an obfuscation mechanism that consists in re-cursively XORing every byte with its previous one, while the first byte of the plain text is simply copied into the obfuscated string. Algorithms 1 and 2 show the pseudo-code of the obfuscation routines visual Encrypt() and visual Decrypt() used by this malware version.

Note that, as a result of Algorithm 1, the unknown terms in a Zeus report header are propagated through the obfuscated string. However, in the following we prove that a *derivate key* γ can be obtained and used to extract the last m elements of any string ciphered using key K, regardless of the application of Algorithm 1.

Definition 1. Let $P = \{\epsilon_0, ..., \epsilon_{n-1}, p_n, ..., p_{n+m-1}\}$ be a string with $|P| = n + m$ composed by a set of n unknown values ϵ_i followed by a set of m known values p_i. Let K be a cipher key and C the cipher text obtained as $C = Obf(P) \oplus K$. We define a *derivate key* associated to C as $\gamma = C \oplus O'$, where O' is the obfuscation of a vector $P' = \{\varphi_0, ..., \varphi_{n-1}, p'_n, ..., p'_{n+m-1}\}$ with $p'_i = p_i$ for $i \in [n...n+m-1]$ and φ_i taking any known value for $i \in [0...n-1]$.

Theorem 1. *Given a cipher text $Z = Obf(Q) \oplus K$ with $|Z| = n+m$, the last m elements of the string Q can be obtained using the derivate key γ associated to C.*

PROOF. The *derivate key* is defined as $\gamma = C \oplus O'$, where the first term can be expressed as

$$C = Obf(P) \oplus K = \begin{cases} Obf(\epsilon_i) \oplus k_i \\ \quad \text{for } i \in [0...n-1] \\ \\ Obf(\epsilon_{n-1}) \oplus p_n \oplus ... \oplus p_i \oplus k_i \\ \quad \text{for } i \in [n...n+m-1] \end{cases} \quad (1)$$

while O' is obtained as follows

$$O' = Obf(P') = \begin{cases} Obf(\varphi_i) \\ \quad \text{for } i \in [0...n-1] \\ \\ Obf(\varphi_{n-1}) \oplus p'_n \oplus ... \oplus p'_i \\ \quad \text{for } i \in [n...n+m-1] \end{cases} \quad (2)$$

Since $p_i = p'_i$, γ can be rewritten as

$$\gamma = \begin{cases} Obf(\epsilon_i) \oplus Obf(\varphi_i) \oplus k_i \\ \quad \text{for } i \in [0...n-1] \\ \\ Obf(\epsilon_{n-1}) \oplus Obf(\varphi_{n-1}) \oplus k_i \\ \quad \text{for } i \in [n...n+m-1] \end{cases} \quad (3)$$

Let us consider an unknown plain text $Q = \{\lambda_0, ..., \lambda_{n-1}, q_n, ..., q_{n+m-1}\}$. The corresponding cipher text Z can be expressed as

$$Z = Obf(Q) \oplus K = \begin{cases} Obf(\lambda_i) \oplus k_i \\ \quad \text{for } i \in [0...n-1] \\ \\ Obf(\lambda_{n-1}) \oplus q_n \oplus ... \oplus q_i \oplus k_i \\ \quad \text{for } i \in [n...n+m-1] \end{cases} \quad (4)$$

When applying the *derivate key* to Z, we obtain

$$Z \oplus \gamma = \begin{cases} \Delta_i \\ \quad \text{for } i \in [0...n-1] \\ \\ \Delta_{n-1} \oplus q_n \oplus ... \oplus q_i \\ \quad \text{for } i \in [n...n+m-1] \end{cases} \quad (5)$$

where $\Delta_i = Obf(\epsilon_i) \oplus Obf(\varphi_i) \oplus Obf(\lambda_i)$. Finally, values q_i can be obtained for $i \in [n...n+m-1]$ by applying $Obf^{-1}(Z \oplus \gamma)$ to eliminate the constant term Δ_{n-1}.

Note that although terms λ_i cannot be obtained, these are not relevant in order to identify malicious traffic.

4.3.2. Key Extraction algorithm

Algorithm 3 Create Obfuscation Set algorithm

```
1:  function CREATEOBFSET(text)                ▷ The chosen plain text
2:      T ← text
3:      x ← 112
4:      for i ← 0, 20 do
5:          insert(T_x, "A")
6:          O'_i ← visualEncrypt(T)
7:          i ← i++
8:      end for
9:      return O'                              ▷ The obfuscated set
10: end function
```

One of the aims of this research was to obtain enough parts of γ needed to decipher the Zeus configuration file, which is usually around 69 KBytes [30]. By executing a Zbot sample in our sandbox, it was possible to build a packet containing *chosen-plaintext* by inflating a modified cookie, i.e. here in after *contrast-cookie*, with ordinary strings. The *contrast-cookie* takes a relevant role because it allows us to enrich our *chosen-plaintext* to obtain a bigger known plain text, which implies being able to consecutively retrieve the desired size of γ. On top of that, we were able to extract the last *m* bytes of the *derivate key* γ and to

Algorithm 4 Key Extraction algorithm

```
1:  function EXTRACTKS(payload_1, payload_2, knownPlain)
2:      P' ← knownPlain
3:      C ← payload_1
4:      D ← payload_2
5:      O' ← createObfSet(P')
6:      for i ← 0, |O'| do
7:          for t ← 0, |O'_i| do
8:              S_i,t ← O'_i,t ⊕ C_t
9:              i ← t++
10:         end for
11:         i ← i++
12:     end for
13:     for i ← 0, |S| do
14:         for t ← 0, |S_i| do
15:             O''_i,t ← S_i,t ⊕ D_t
16:             i ← t++
17:         end for
18:         E_i ← calcEntropy(O''_i)
19:         i ← i++
20:     end for
21:     v ← calcMin(E)
22:     γ ← S[v]
23:     return γ
24: end function
```

decipher the configuration file without knowing neither the RC4 keystream *K* nor its *seed*. In addition, the pro-posed approach would also save the researchers from carrying out the static malware analysis to search either the keystream or the seed in the system volatile memory.

Note that, according to Theorem 1, the *derivate key* γ can be applied to any string *Z* obtained when ciphering a string *Q* formed by a set of *n* unknown values and *m* known ones with key *K*; however, in order to obtain γ, it is necessary to establish a correspondence between *pi* and p'_i, which requires to know *n* in advance. As stated in Section 3, a Zeus *report* contains two unknown substrings: the Zeus *header* and the Zeus *trojan item* 10002, i.e. the *botnet-id*. Although the length of the Zeus *header* is known, this is not the case for the Zeus *botnet-id*. Given that the mentioned field has a limited length, we tackle with this issue by performing a *brute-force attack* on the Zeus *botnet-id item body* length.

According to the Zeus packet structure explained before, the length of every Zeus *item body* is declared in 4 bytes of its *header*. Thus, the maximum length of the field is 2^{32} bytes long. However, the Zeus Builder toolkit does not allow to create malwares containing a *botnet-id* value bigger than 20 characters. Therefore, we just need to make an attempt over 20 di*ff*erent positions as for where to insert our known *environ-mental item* values. Algorithm 3 is in charge of creating the set of all the possible obfuscated texts O', depending on their different lengths. Note that in line 5 the insert function is used for adding the character "*A*" at the fixed position[2] *x*, and shifting all the next values by one byte in the string. The algorithm takes the *chosen-plaintext* as input, and returns a multimodal-array with all the related obfuscated texts.

Once we generate our set of O', we can use it for retrieving the *derivate key* as explained before. The key extraction pseudo code is expressed in Algorithm 4.

- **Lines 1-4** The function extractKS() is called with three arguments: the intercepted payload *C*, the *chosen-plain text* P' previously generated, as well as a second payload
  D, corresponding to another infected computer belonging to the same botnet.

[2] This position depends on the length of the Zeus *item* id 10001

- **Line 5** The set of obfuscated texts *O'* is created in order to test all possible lengths of Zeus' *botnet-id item*.
- **Lines 6-12** A *derivate keys* matrix *S* is generated as explained before, where every element S_i corresponds to one possible Zeus *botnet-id item* length.
- **Lines 13-20** In order to discover the appropriate *derivate key*, every key *Si* is tested against the second payload *D*. As a result, a matrix *O"* containing all the possible decryptions of *D* is obtained. Moreover, the entropy of each possible decryption is computed in line 18.
- **Lines 21- 24** The appropriate derivate key Sv = γ is identified due to the lower entropy Ei resulting of its application to the payload D.

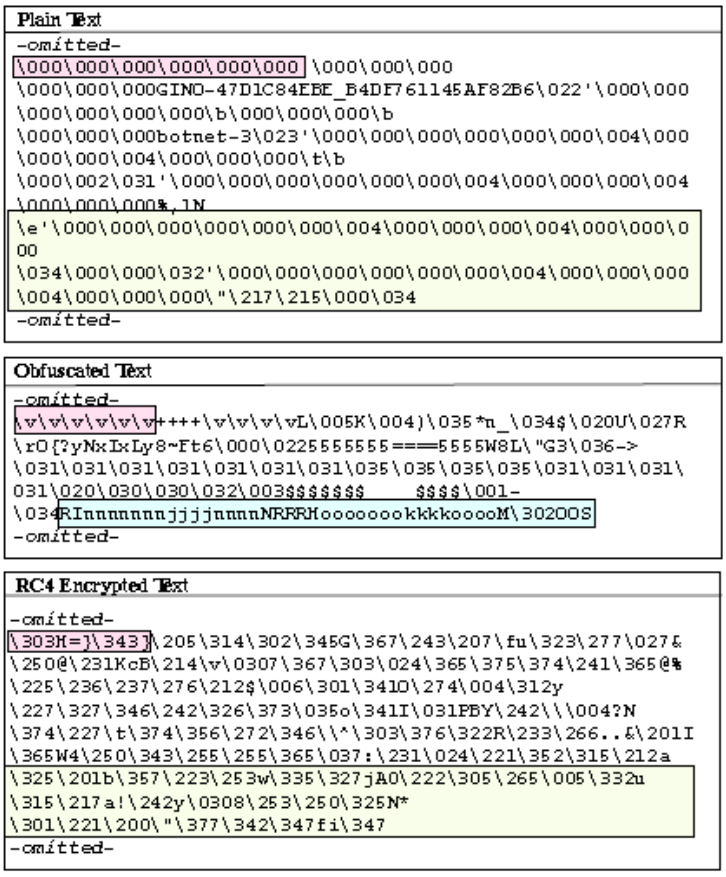


Figure 3: Zeus data through the three deciphering phases

Once a reasonable part of γ is successfully extracted, it is sent to *Cronus* in order to let it be able to identify Zeus tra*ffi*c by performing *deep packet inspection*.

4.3.3 Cronus: an IDSfor Zeus

Our research further demonstrates how the entropy of these obfuscated messages *O* remains nearly constant, allowing us to automatically identify malicious tra*ffi*c even if it is not fully de obfuscated. This propriety derives from the *zero padding strings* used by Zeus for composing its messages.

As can be seen in Figure 3, when these strings are obfuscated by the `Cronus()` routine, the resulting output contains several text parts with repeated characters (these parts are evidenced in the cited figure). As a result, the calculated entropy drastically de-creases in comparison with the one calculated for the cipher text.

Algorithm 5 illustrates the pseudo code of *Cronus*, the pro-posed Zeus IDS. The algorithm works as follows: a) once the suspicious HTTP payload has been filtered out, it is passed as an argument to the `makeitPlain()` function. b) The Shannon en-tropy H of the message is then calculated and, if it is higher than the threshold α, it means that the analyzed payload C is encrypted. Hence, the algorithm can continue its flow[3]. c)

Bytes range	Content	Binary values
48 – 51	\021'\000\000	1001
52 – 55	\000\000\000\000	0
56 – 59	\000\000\000	32
59 – 63	\000\000\000	32

Table 2: Zeus ItemID 1001 header

Zeus message segment	Bytes range	n. of *null bytes*
Zeus *header info*	20 – 31	5
Zeus *item* 1001 *header*	48 – 63	12

Table 3: Pattern used by *Cronus*

The function continues with the `RC4()` routine which XOR-decrypts C with all the γ values previously collected and, for each decrypted output O, the routine `calcentropy()` is called to calculate its entropy H'. d) If H' is lower than β, it means that O is likely to be a Zeus obfuscated message. As said, there is no need to deobfuscate O in order to confirm the detection.

It is important to note that the condition $|\gamma i| \geq |C|$ should be verified before calling the `Cronus` function. Because of this, we are able to identify only those packets with cardinality less or equal than the collected key one, meaning that the key should be big enough (300 Bytes at least) to correctly

[3] Note that from Information Theory we know that the Shannon entropy is not increased by more than the length of the encryption key. However, dealing with RC4 encryption, the encryption key is as long as the plaintext is. There-fore, our choice of the Shannon entropy to discriminate whether a message is encrypted is meaningful.

decrypt a Zeus *log* message and determine its maliciousness. By considering that this could raise a performance issue for *Cronus* due to collecting several keys and use them to analyze each HTTP POST packet, we focused on only searching for short and delimited parts of Zeus packets and analyze them accordingly.

We found two different patterns that, once encrypted and obfuscated, always maintain certain contents at the same position, and keep their entropy values constant. In order to find a pattern useful to identify a Zeus traffic, we need to use the *static* parts of its messages with contents changing as less as possible, and that are always located at the same position of the Zeus message. Given these constraints, we focused on the 12 bytes reserved for the Zeus *header info*, and the 16 bytes reserved for the first Zeus *item header*. While other *item header*'s positions are variable, i.e. depend on the size of the previous *item body*, those headers can always be found at the same position of the TCP packet's payload, i.e. between byte 20 and byte 31, and between byte 48 and byte 63 respectively.

Var	Entropy value	Explanation
a	~7.3	Encrypted text
/3	6.0-6.6	Obfuscated text
8	~4.8	Plain text

Table 4: Entropy thresholds used during the experimental phase

Pattern 1: *Zeus header info* – The 12 bytes long Zeus *header info* contains information about the Zeus body size, the item flags, and the number of items. The body size of a typical Zeus log message is around 300 bytes, thus, by considering that the maximum number of bits used to represent this number is 17, and that four bytes are used for encoding the Zeus body size, in the worst case one entire byte will always be set to *zero*. Next, the four bytes long Zeus *item flags* follow a certain structure: the first byte is used as a binary selector in order to inform if the packet is compressed or not, the second byte is used for the instructing the control panel about how to manage the Zeus *items* if they are encrypted, while the third byte is used for reducing the risk of item overlapping. Finally, the last byte is never used, and is always set to zero. Although we can definitely assert that in the worst case one byte will be always set to *null*, it is interesting to note that all the Zeus *log* messages that we analyzed during our research, always kept these four bytes to zero. Lastly, by considering that in the Zeus control

panel source code only 45 different items are encoded, only one byte of the four allocated will be enough to represent the number of the items contained inside a Zeus message.

Algorithm 5 Cronus algorithm

```
1:  function CRONUS(payload)                    ▷ HTTP POST request
2:      H← calcEntropy(payload)
3:      if H ≥ α then                           ▷ The payload is encrypted
4:          for i ← 1, sizeof Γ do
5:              C ← payload
6:              O ← RC4(C, γ_i)
7:              H' ← calcEntropy(O)
8:              if H' ≤ β then                  ▷ Zeus obfuscated data
9:                  return 1                    ▷ Zeus obfuscated data
10:             else
11:                 return 0                    ▷ data not decrypted
12:             end if
13:             i ← i++
14:         end for
15:     end if
16: end function
```

Pattern 2: *Zeus item header*– Recalling the Zeus packet structure exposed in Section 3, the 16 bytes long Zeus *item header* contains the *itemID*, four *null bytes*, and an identical pair of four bytes containing the *item body*'s length. Therefore, as we have already showed, 2^{32} bits are used to express the item's length in bytes. The *botID* is composed of the infected computer's Net-Bios name, which can be 15 byte length at maximum, and a 17 bytes long string randomly generated by the malware in order to assure a unique *botID*. Hence, the Zeus *item body's* length can be 32 bytes at maximum and only one of the four allocated bytes may be used to express this information, while the other three bytes will always contain zero values. As showed in Table 2, the first Zeus *item header* contains several *null bytes*.

Table 3 summarizes the pattern values by highlighting the number of *null bytes* present in the worst case. Due to their nature, these headers always contain several zero strings that drastically decrease the string entropy value. Notably, by using more bytes than required, the malware coder compromised its own obfuscation algorithm by making its output easily detectable. Based on these findings, our research demonstrates that the proposed IDS can identify certain Zeus traffic with only 36 bytes long keys. In this way, the performance of *Cronus* storing and analyzing all the captured HTTP POST requests significantly increases in comparison to storing 1500 bytes long keys.

Finally, once the Zeus bot communication identified, several further actions could be taken. For instance, the traffic could be blocked by a network firewall. This can be easily achieved with *Cronus* sending the information needed to update the firewall rules in order to deny all the traffic directed to the Zeus C&C. The α and $/3$ entropy thresholds used during our experiments are shown in Table 4. These thresholds have been set after a learning phase which included calculating the entropy of 100 Zeus plain texts with their corresponding obfuscated and encrypted transformation. Note that these values are completely customizable.

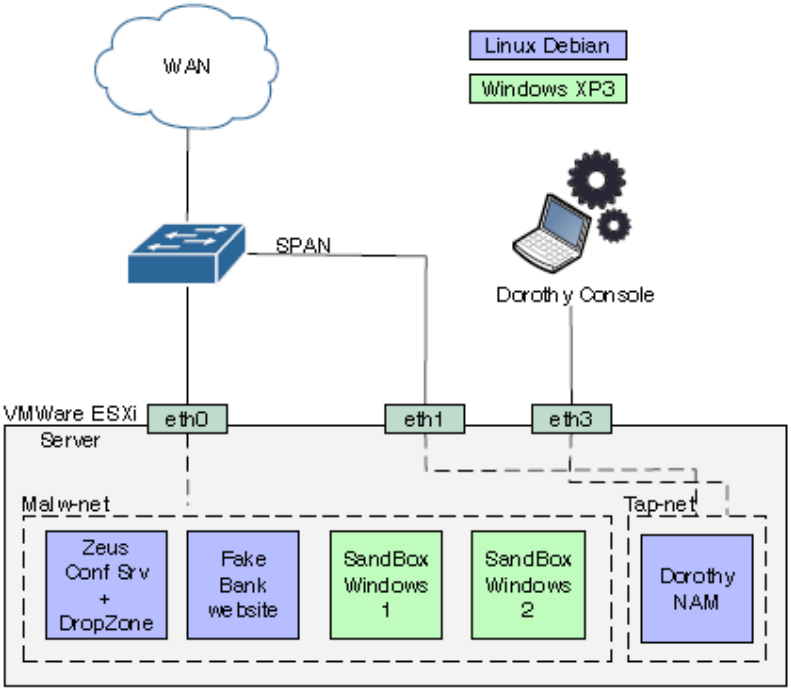


Figure 4: Test environment used during the experiment

4.4. Experimental Settings

The different phases of our experiment are described in the following section. In order to test *Cronus*, we set up a testing environment, which comprises an in-house Zeus botnet that includes a C&C, a *dropzone*, two computers to infect (*v*1 and *v*2), and a web server where a fake banking webpage is running. Once the testing environment was configured, we created a Zeus trojan binary and executed it in *v*1 and *v*2. Furthermore, we analyzed the generated network traffic in order to extract certain parts of the *derivate key*. Finally, we used those keys to develop *Cronus*, an IDS able to detect Zeus traffic in a production network.

4.4.1. Set up of the testing environment In the first phase, we recreated a complete Zeus environment in an isolated network. This task has been accomplished by customizing the *Dorothy framework* [11] that is currently employed for similar research initiatives. Such a framework is built upon a

VMWare ESXi infrastructure that is automatically piloted trough the *Dorothy Console*, i.e. a Ruby gem installed on an external machine which is in charge to start, stop and revert the virtual machines, besides running executables inside them once transferred. The framework comes with a network analysis module (*NAM*) which relies on a *pcapr-local* [31] in-stance that is dedicated to dissect and store the analyzed traffic into a non-SQL database for allowing fast and indexed searches. Such module is in charge of recording and dissecting all the net-work traffic generated by the sandboxes for the whole execution of the VMs. , we configured four different virtual machines: a Linux Debian distribution was used for configuring the Zeus control panel and its *dropzone* and two Windows virtual machines were selected as sandboxes for the malware execution. Finally, another Linux-based VM was used for running a fake bank website. The Windows sandboxes come with a Windows XP SP3 default installation, with Internet

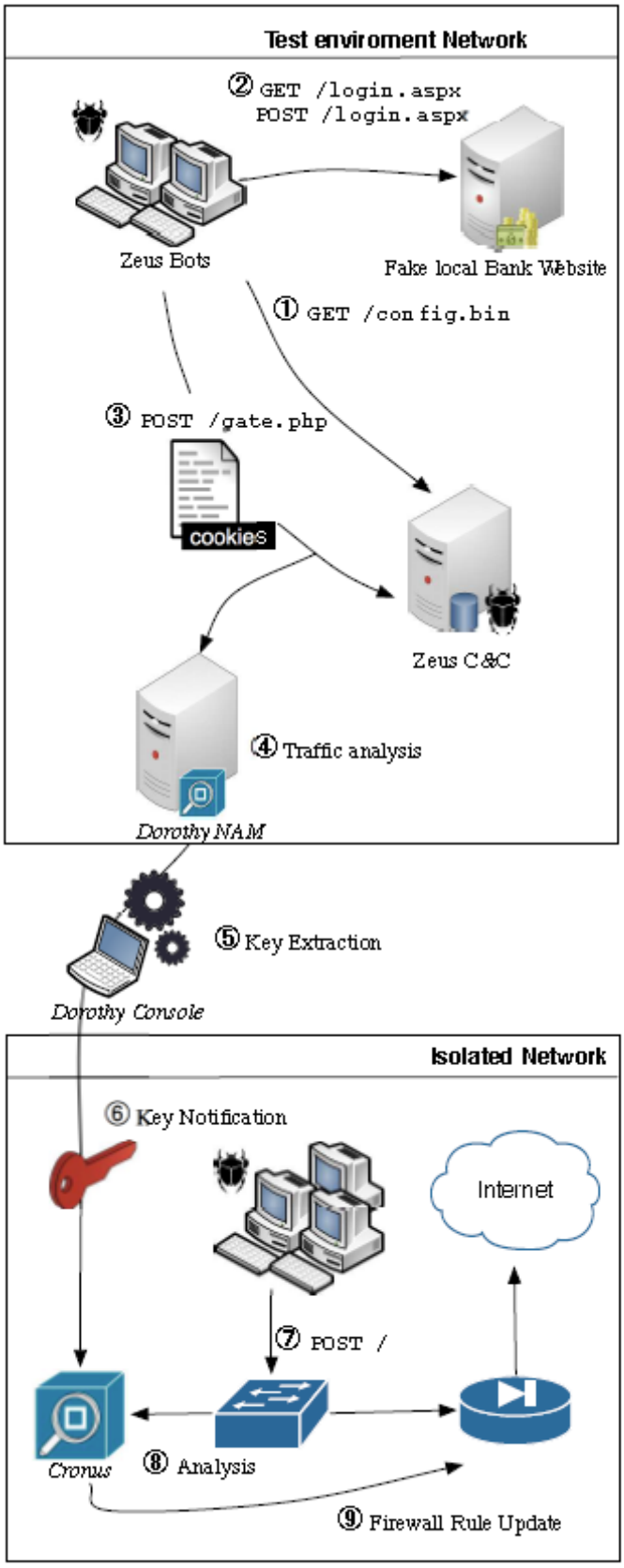


Figure 5: Analysis Flow

Explorer 7 installed on it. In addition, a *contrast-cookie*, associated to the local fake bank website was inserted in the default IE system cookies folder /Documents and Settings/userhome/Cookies of both sandboxes, and modified in order to contain a big quan-tity of repeated strings (we tested a *contrast-cookie* up to 700 Kbyte large). The fake web server ran on a TomCat server, which replicated the login page of a known home banking web-site. The local DNS server was configured consequentially in order to resolve the real bank site URL with the local IP address. A summarized scheme of the test environment is presented in Figure 4.

Finally, we used the provided Zeus Builder to create a new malware sample *mw*1 and its relative configuration file *conf*1 and moved them to the sandboxes and to the *dropzone* respectively. The configuration file contains the local IP addresses of our testing environment as *dropzone* and C&C. Through this file, we configured our bots *v*1 and *v*2 in order to communicate with the C&C every two minutes, and to send the full system *reports* every four minutes. Notably, these *reports* contain all the system cookies, including our inflated *contrast-cookie*.

4.4.2. Malware execution and Key extraction

The whole malware analysis process is described in Figure 5. The process begins by transferring *mw*1 to the sandboxes, and executing it with administrative privileges. We define the *zombified timeframe* At_e as the interval A*t* between the time of the malware execution and the time when the virtual machines *v*1 and *v*2 are reverted to their not malicious activities.

During At_e, the *NAM* records all network traffic generated by the infected VMs, and stores it in a PCAP file. At the early age of its infection, the infected systems attempt to retrieve the encrypted configuration file by making HTTP GET requests to the Zeus C&C (Step 1). Notably, the C&C IP address/domain name is hard coded inside the malware binary, and it could be revealed by conducting a binary static analysis. Just after the execution of the malware, both sandboxes are automatically driven to visit the fake bank website, and to enter fake login credentials (Step 2). Web automation is accomplished by executing a customized macro of the iMacros plugin [35], which successfully emulates the usual human internet

browsing activity.

After *con f*1 is downloaded, each infected system deciphers it and extracts the information related to the botnet *dropzone* do-main name/IP address. At this point, the bot begins to make two different types of HTTP POST requests to the C&C (Step 3), i.e. *Zeus logs* and *reports*. Differently from the first type of request, the size of the data transmitted in the Zeus *report* depends on the information stored in the infected computer. As previously outlined, we leveraged this propriety to inflate our *contrast-cookie* in order to extract enough parts of the keystream needed to further decrypt *con f*1. An example of a decrypted report message (where relevant parts are highlighted), can be seen in Figure 6.

Figure 6: Information sent to Zeus dropzone

After At_e, the sandbox virtual machines are reverted to their original state, and the saved network dump is analyzed inside the *NAM* by dissecting it and extracting all the network flows recorded during the analyzed time (Step 4). Consecutively, the *Dorothy Console* analyzes them by filtering out only those flows related to HTTP traffic directed to the local Zeus C&C, and then extracts the payload of the largest ones, which supposedly contain the stolen cookies of *v*1 and *v*2. Once the ciphered flows have been filtered out, they are used for extracting the *derivate key* as explained in Section 4 (Step 5).

Finally, the returned keystream is used for decrypting the encrypted configuration file, and if the retrieved output gives acceptable entropy ($H' \leq$ ¡3), then the tuple < *mwi*, γ*i* > can be sent to the *Cronus* database (Step 6). Notably, this operation is accomplished by the *Dorothy Console* which connects the *NAM* with the production network, e.g. where *Cronus* resides.

4.4.3. Malicious traffic identification

The last phase consists of detecting certain Zeus traffic on a production network, having a set of valid Zeus keystreams Γ = **{**γ0, γ1, ..., γ*i***}**, obtained as detailed in previous subsection. To perform this test, we set up five different virtual machines *V* = **{***v*0, ..., *v*4**}** in an isolated network (Step 7), which constantly generate common network traffic towards a fixed hosts range *S* = **{***s*0, *s*2, ..., *si***}**, e.g. web browsing and POP mail consulting. Next, we infected one of them with a previously created Zeus malware sample *mw*1, and recorded all network traffic generated by this machine during At_e (Step 7). Another virtual machine is configured as network sniffer in order to analyze all network traffic going through the network. This latter VM also executes our Proof of Concept (PoC) for *Cronus*, which is designed to scrub all HTTP POST requests identified in the net-work flow (Step 8), and trigger an alert whenever a malicious pattern is detected (Step 9). It is relevant to note that the net-work flows analyzed by *Cronus* discarded HTTPS traffic, other-wise the algorithm would waste time and resources by trying to decrypt a non-RC4 traffic. Additionally, other filtering proprieties, e.g. white lists, can be added to reduce false positives and to avoid unnecessary load charge on *Cronus* that can be generated by analyzing legitimate traffic. Finally, if *Cronus* returns positive results for an analyzed network flow, we can claim that *vi* has been infected by *mw*1, and that *si* is a specific Zeus C&C which has been proved to be online during during At_e.

The *Cronus* PoC was developed to decipher all the suspicious traffic identified in the network traffic. Since the Zeus botnet is based on the HTTP protocol, and the stolen information is sent to the *dropzone* through HTTP POST requests, the following is recognized as *suspicious* traffic and analyzed by *Cronus*:

1. TCP traffic;

2. An HTTP POST request containing an encrypted body.

While filtering a TCP network stream is quite easy, filtering HTTP POST requests requires analyzing the upper lever of the TCP protocol. This technique is commonly referred as *deep packet inspection* and requires a fast CPU in order to process the network flow in reasonable times. The proposed *Cronus* PoC was developed in Ruby language, and can be deployed and executed on any system with the Ruby Framework installed on it.

Next section reports on the results of our experiment.

4.4. Evaluation of the results

In this section we describe the applicability of the proposed approach inside a corporate network during a regular working day. The goal of this experiment was to demonstrate that the proposed approach could be used in a real scenario; by offering to the network administrator the capability to detect Zeus infected systems.

4.1. Environment configuration

As first step we created a customized Zeus Trojan V. 2.0.8.9 by using the toolkit leaked on Internet. The malware was con-figured in order to communicate to our internal server where a Zeus control panel was previously set up. In addition, we de-fined an interval of 2 minutes for the Zeus *log* messages, and 10 minutes for the *report* ones. The experiment began by executing an instance of `tcpdump` in a machine which was physically connected to a SPAN port of the corporate router. In order to limit the impact of the experiment, the SPAN port was configured in order to redirect only the network traffic belonging to a VLAN composed by 13 host, i.e. 11 desktop computers used by researchers for their daily job, and 2 virtual machines dedicated for the experiment's purpose. *Cronus* was executed in a Linux Debian Etch machine, with 2Gb of dedicated memory and a 2.6Ghz CPU.

The sniffer began to record the local traffic at 9:50 AM , and lasted its activity for the next 5 hours. At 11:35 AM, a Windows XP SP3 virtual machine inside the same network was infected with the previously generated malware, and an instance of iMacros was configured to mimic a typical web browsing activity by executing a macro which starts from the Google News web page and browses its link every 10 minutes. At 3:00 PM, the `tcpdump` instance was stopped, and an overall of 17.50 GB of network traffic were correctly collected.

The malware keystream was previously extracted by using the explained approach, and it was stored into a text file which was already containing 28 different keys. In addition, 200 more different keys were inserted into the key file, in order to assess the computational overhead experienced by *Cronus* during the keys loading function[7].

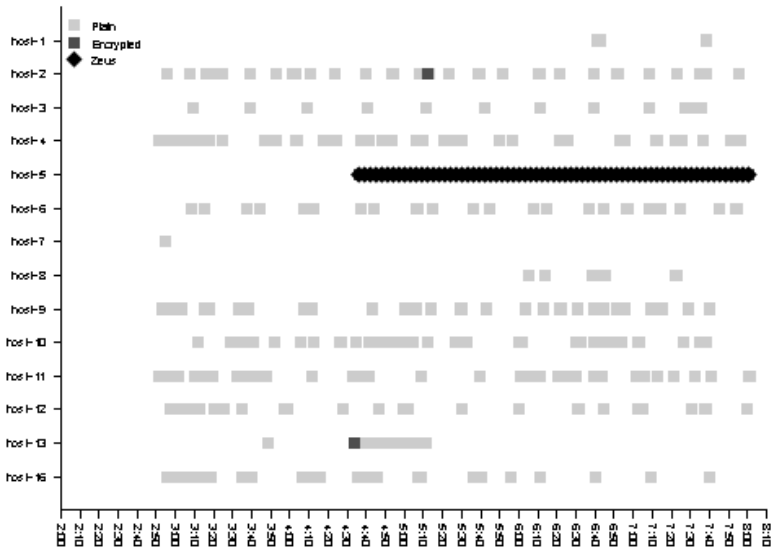


Figure 7: Analyzed HTTP POST requests

Once the network dump was generated, it was moved to the *Cronus* machine. *Cronus* was configured to load a file containing 229 different keys but to test only the first 30 in order to simulate a real case scenario[4]

6.2. Results

Cronus analyzed 7,133,170 TCP packets finding 5,990 HTTP POST requests, 141 of them containing an encrypted payload. For each of these, 30 different keys were tried in order to de-cipher the payload and retrieve a reasonable entropy result. As result, 138 HTTP POST requests were correctly identified as Zeus traffic.

[4] [8]We decided to test only 30 keys because at the time of this writing, there are only 26 Zeus active domains standing to Zeus Tracker[30].

Figure 7 shows that the three missing encrypted POST re-quests didn't belong to the infected machine (`host-5`), thus are not false negatives. However, after a manual analysis of the packets sent by the infected machine, we discovered that it generated 139 HTTP POST packets toward the Zeus C&C. The packet which was not detected by *Cronus* belonged to the Zeus *report* communication flow, which contains all the cookies of the infected system. As explained before, this packet is typically larger than the network MTU, and packet fragmentation is often needed to correctly send it. As a matter of fact, the TCP segment was 5785 bytes large, and it was splited in 9 frames. The first frame, contained only the HTTP POST header, and it was correctly detected by *Cronus*. However, the following ones were not detected because they did not contain the *POST* keyword inside their body — as showed, *Cronus* only considers these packets for its purpose. It is important to note that big Zeus *report* packets are definitely less frequent than the *log* ones. Hence, taking into account *packet reassembling* would not sensibly contribute to enhance the detection rate, while drastically decreasing the analysis performance.

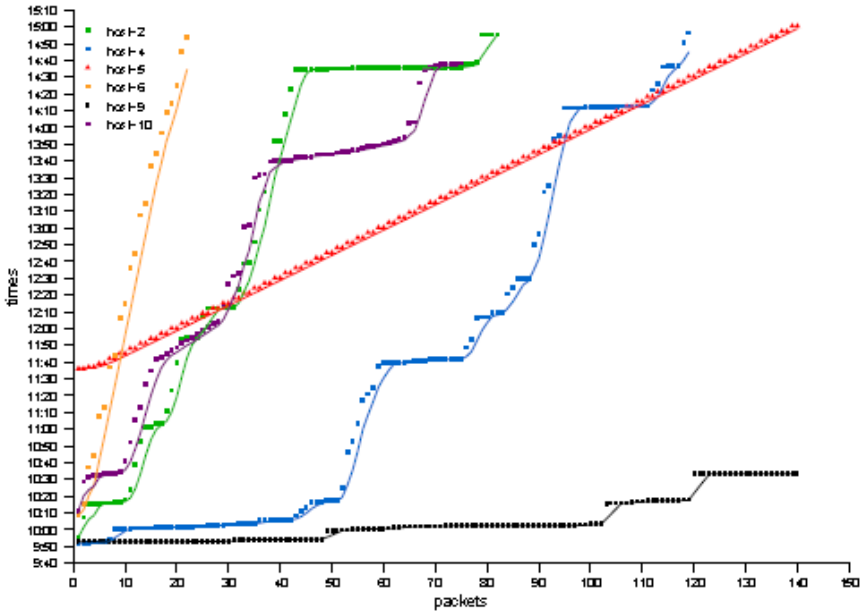


Figure 8: HTTP POST requests per host

Figure 8 highlights the frequency of all the HTTP POST re-quests sent by the most active hosts. Note that `host-5` has a constant request rate, that identifies the typical Zeus *log message* activity. Interestingly, this graph could help a system administrator to identify suspicious Zeus traffic by only mapping over time the host HTTP POST requests.

The time of the analysis was of 5.39 minutes, which means that the estimated throughput of the proposed IDS is around 400 Mbit/s. The performance of *Cronus* strictly depends on the number of the keys tested, and on the network dump file size. However, as showed in Figure 9, the number of tested keys does not drastically impact on the execution time[5] if we consider that there are less than 90 different Zeus botnet spotted in the wild. However, as the file size grows, the execution time increases as well, according to a linear relationship.

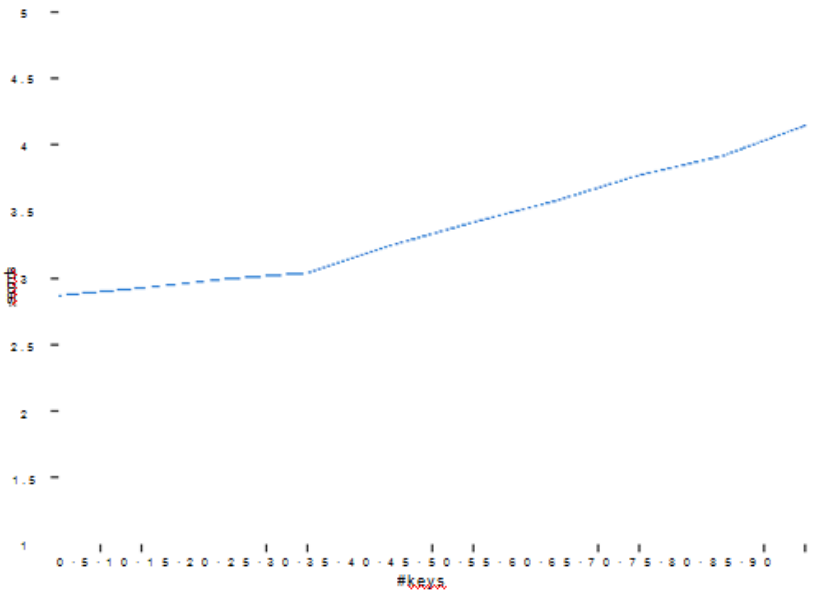


Figure 9: Execution time per number of used keys

4.5. Lessons learnt

One of the main contributions of this paper, apart from the detailed techniques outlined so far, is the fact that we have high-lighted a general methodology to attack a class of malware that can be considered generic enough to account for a reasonable part of the total amount of malware posing a serious threat to the Internet. The approach can be essentially decomposed into the following steps: 1. Isolating crypto modules of the malware from the rest of the application; 2. analyzing the crypto modules as for their operating procedures; 3. discovering flaws in the crypto modules. Note that the weaknesses we have been investigating concern the composition of the crypto module with the rest of the application; we did not focus on breaking the crypto algorithm. In our investigation, we have not devoted re-sources to make an attempt to break the RC4 implementation the crypto module is based upon. Instead, with the above high-lighted approach, we have discovered two weaknesses: a. the crypto module is subject to a re-

[5]The execution time values has been calculated by executing *Cronus* on a16,5 Mb large PCAP.

initialization attack; b. the plain text to be encrypted can be provided in input to the algorithm. These two weaknesses, combined with the fact that the onetime pad encryption (based on the keystream generated by the RC4) is subject to the cancelation property, have paved the way to our attack in recovering a relevant portion of the keystream. It may be legitimate to think that another weakness is represented by the short length of the *botnet-id* field. Although a bigger *botnet-id* would certainly impact on the keystream's recovered fragments, it would also imply several drawbacks for the C&C server management: this unique field is used to update botnet information for every incoming packet, thus doing a MySQL distinct query by filtering a — let us suppose— 2^8 bytes long *botnet-id* would be cumbersome in terms of system resources. In addition, Zeus *log* communications length between the bot and its C&C, would drastically increase, hence making them easier to detect[6].

The sequel of our technique and proposed architecture has just been an exercise of security engineering and secure net-work design. Interestingly, the attack we proposed is just a reviewed version of the Crib-based one used at Bletchley Park while breaking the Enigma crypto system [35]: the known Zeus *environmental items* represented our *cribs* and the proposed key extraction algorithms our *bombe*. We can definitely assert that old-fashion sound cryptanalysis still works, even 70 years later.

Finally, it is important to note that while the proposed technique is unique to Zeus malware versions prior to 2.0.8.9 and to some later variants that still continue to use the same crypto routine, trends of malware flaws in the implementation and use of crypto algorithms (like the exploited one) could be potentially used as directions for reverse engineering of other malware in the wild.

[6] Reader who might be interested in finding out more about the counter-measures which could be used to avoid the exposed detection mechanism may contact us. As a result of an internal discussion it was decided to not disclose those techniques in this paper to avoid it being used by malware developers.

5. CHAPTER II: APP and mobile Malware

Mobile devices, such as the smartphones and tablets are playing an increasingly important role in how people communicate, socialize, and carry out other important daily tasks including accessing their banking activities. Newer network technologies as well as the mobile devices will progressively increase security aspects that would contribute to the wider range of mobile applications and services.

Mobile applications stores (App-Stores) are no exception and the recent popularity of different mobile App-Stores such as the Google's Android Market, and Apple's App-Store, is a clear evidence to support the above statement.

According to an article by Minda Zetlin (Zetlin, 2011) [37], a number of Android users, in the year 2010, downloaded different mobile banking applications (Apps) from the Google's Android Market at a cost of $1.50 each. The Apps enabled the users to connect with about 40 major banks, including the Bank of America and the Wells Fargo of the United States. It later appeared that the banks did not upload those Apps and that those Apps were in fact submitted by some unknown fraudsters to different App-Stores, seeking to only make $1.50 from each download. Needless to mention the potential threat of the fraudsters being able to steal the

users' banking login credentials, as a result of the users downloading those malicious banking Apps. It was also the reason why many banks asked their customers to actually have their mobile service provider remove those malicious Apps from their mobile devices (Zetlin, 2011). [37]

Another work (Goodin, 2011) stated that the Google's Android Market had to remove at least a dozen smartphone games out of the Android Market after discovering they contained secret code that was sending text messages (SMS) to a premium number and the users having to bear the high costs of those text messages (Goodin, 2011) [36].

Moving on, the recent evolution within the mobile banking ecosystem clearly indicates that the access to banking through mobile devices will move from browsers to mobile Apps. Moreover, a great part of the security

will depend on how secure is the actual App-Store as that is where a user would download their banking App from, in order to access their mobile banking and other financial activities. It is therefore vital to address these growing malicious Apps threats and one of the measures could include outlining some sort of common policies for different App-Stores, enabling them to better manage their Apps delivery mechanisms.

In this work research, we aim to outline a proposal on common policies for mobile Apps management mechanisms for different mobile App-Stores. Therefore, an assessment of the major mobile App-Stores was carried out in order to establish an understanding of their current Apps delivery/monitoring mechanisms. This included the current procedures involved in uploading, downloading, and upgrading the Apps to/from an App-Store along with the current reporting mechanisms deployed by different App-Stores for the malicious Apps reporting.

5.1 A discussion on different mobile APP-Stores Applications Management Mechanisms.

There are a number of unique challenges and pieces to application security that most devices currently, are not able to provide. Similarly, all major mobile platforms bear different Apps delivery mechanisms and hence the associated risks also differ to a certain extent. One of the common major tasks however, for all mobile Apps platforms/providers is to ensure the Apps authenticity and a secure management/delivery mechanism for all Apps available on their App-Stores.

The differences between the capabilities of platforms have a significant impact on the management of their security. Managing the Apps delivery mechanism is all about the capabilities of a mobile devices platform/App-Store to monitor/authenticate the legality of an App’s source. Being able to keep a track/record of where an App came from can be vital to deploy appropriate countermeasures against a malicious or a fraudulent App.

Furthermore, the rapid changes in this largely consumer driven mobile devices market mean that code is quickly written, deployed and replaced or even upgraded. Development platforms that support the writing of a secure code are currently lacking for the mobile devices. This is particularly the case for the mobile devices Operating Systems (OS)

which are often written in either "C" or other native languages leaving security totally at the discretion of the developer.

Therefore, balancing the restrictions imposed by application delivery mechanisms while ensuring an acceptable level of versatility and usability of the smart phone is a challenge for the Apps providers/vendors. Some vendors provide for encoded signatures on applications and some restrict applications to a single controlled source, while others have no restrictions on the source of an application. Table 1 below, outlines the Apps delivery security mechanisms which are put in place by different mobile platforms:

Table 1 - Outlines the Apps Signing, Revocation, and Approval Procedures for Different Mobile App-stores (Source: Veracode[1])

Platform	**Signing**	**Revocation**	**Approval**
Symbian	Signed by Vendor	Yes	Quality
Android	Anonymous, self-signed	Yes	No
iOS	Signed by Vendor	Yes	Policy and
Windows	Signed by Vendor	Yes	Policy,
Blackberry	Signed with Vendor	Yes	No

http://info.veracode.com/Whitepaper-2011 1

As it can be gathered from the above table, all mobile platforms have some sort of App signing mechanism in place. Similarly, it is also apparent that all mobile platforms support "revocation" in order to remove malicious Apps, once reported or detected.

Figure 1. Android applications development and publishing process[2]

Moving on, the mobile OS will not allow Apps that are not signed to execute with an exception of the "jail-broken" or "rooted" OS's which are in fact "jail-broken or rooted" by the users specifically to allow unsigned apps to be executed.

Depending on the implementation of the signing mechanism, it can be a great achievement in terms of improved security. For instance, if the App is signed by the developer with a self-generated key, there is little security gain but if the application is signed by a key issued by the platform provider then there will be a security benefit based on the policies the platform provider adheres to, for approving Apps. Moreover, it is important to stress that the mobile jail-breaking removes the security benefits of the platform signing mechanism altogether.

Android's Market App store for instance, is probably not up to the challenge when it comes to malicious Apps publishing and distribution. This is evident in the ease with which malicious Apps can be uploaded and distributed on the Android Market.

Fig.1 outlines the process of applications development and publishing on the Android Market: Those malicious Apps can access the sensitive OS resources such as the text messages, mobile device location via GPS, camera, voice recording, to name a few. It can therefore be stated that developing, publishing and distributing a powerful fraudulent Android App which could steal one's personal information including their financial information is almost trivial as the current security measures deployed around the App submission process by Android are inadequate to identify and prevent submission of malicious applications to the Android Market.

The Apple's App Store on the other hand, appears to be a "walled garden" and it does employ an approval process but the details of its approval process are not well documented or publicized. It is therefore difficult to establish exactly what sort of details the iOS Security Team at the Apple App store takes into consideration, when it comes to an App approval or screening. Having said that, it is quite clear, based on the deployment of

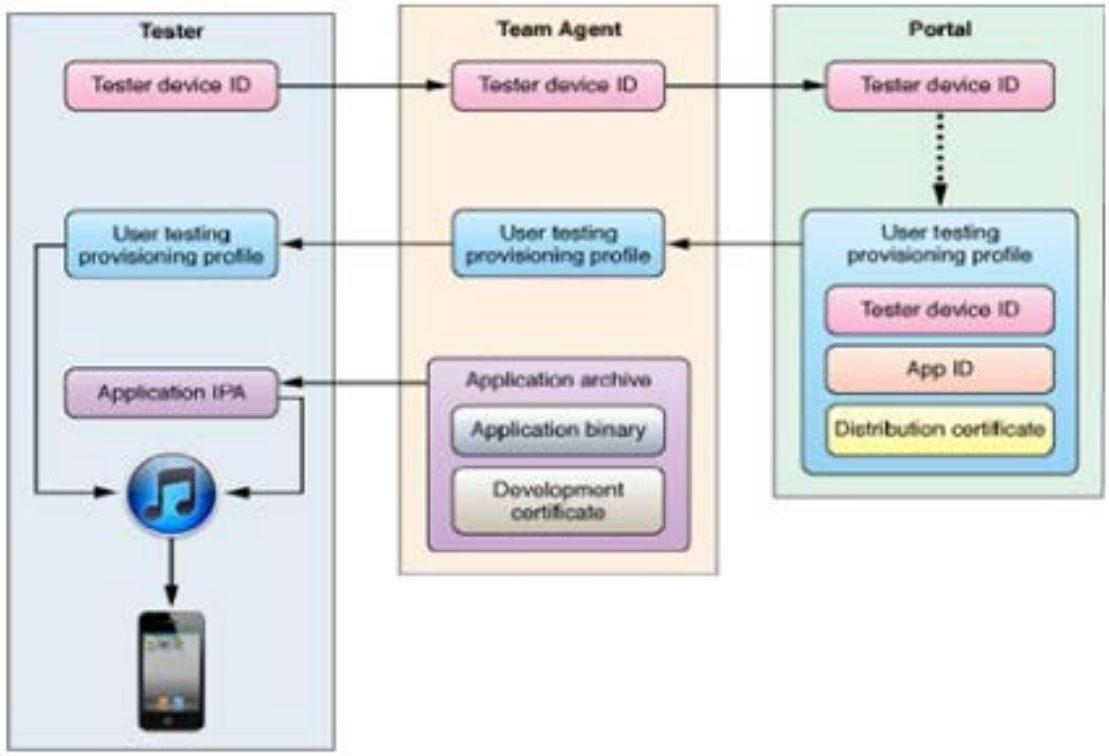


Figure 2. A process of sharing Apple's mobile Apps with testers[3]

the aforementioned approval process, that the Apple App store is putting a lot of effort in ensuring the user experiences and the compliant to its policies.

Fig.2 outlines a procedure of sharing an Apple's mobile App with tester: It is therefore difficult to establish exactly what sort of details the iOS Security Team at the Apple App store takes into consideration, when it comes to an App approval or screening. Having said that, it is quite clear, based on the deployment of the aforementioned approval process, that the Apple App store is putting a lot of effort in ensuring the user experiences and the compliant to its policies.

3 http://developer.apple.com/library/ios 1

According to a recent report from Symantec[7], Apple and Google are very different when it comes to mobile security, “creating distinct potential vulnerabilities for enterprises embracing devices running these operating systems". Apple employs an “Application Provenance” strategy which basically involves identifying, certifying and vetting an App before it is published on to the Apple App store. For Android Market on the other hand, the course of action is completely different as there is no vetting process as there are far more self-signed applications and the Apps can be uploaded from just about anywhere on the internet. So although these user-friendly Apps are continuously getting more and more popular

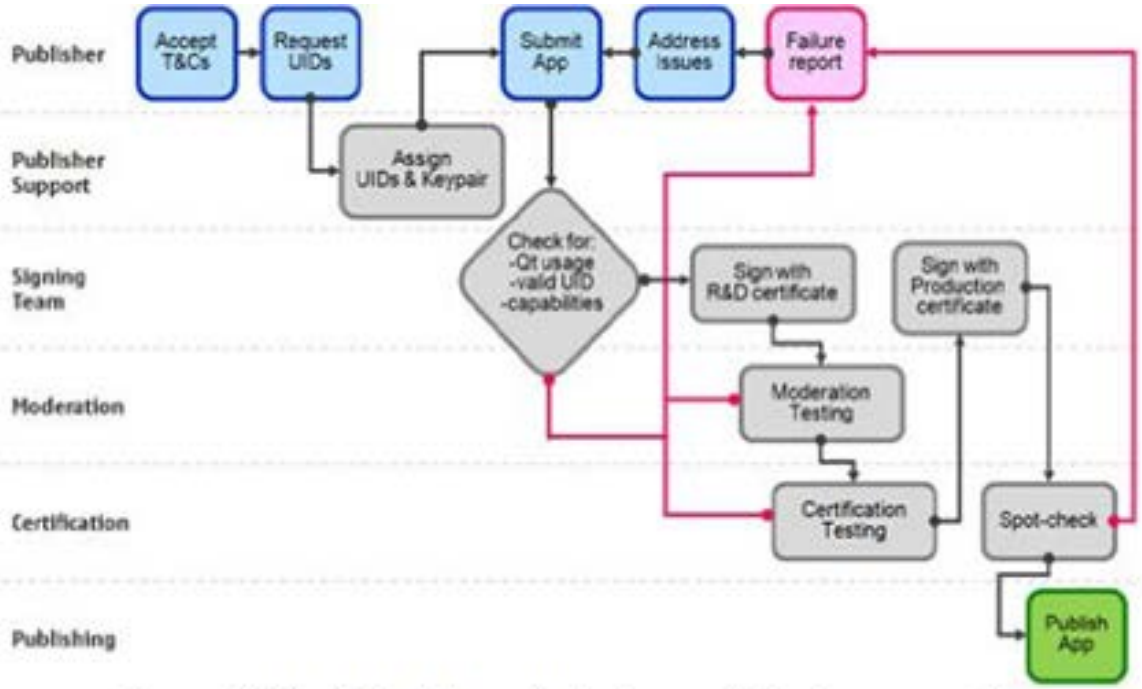


Figure 3 -Symbian App submission and signing process[5]

amongst the users, these are also bringing new security threats and breaches with them.

One of the proposed solutions could be that the Android apps are not only self-signed but are also signed by certified keys issued by the trusted authorities. This strategy on its own may not prevent malicious Apps but it would certainly assist in deploying a tracing mechanism which would allow tracking down the fraudulent App owner/publisher.

[7] http://www.networkworld.com/news/2011/062811-symantec-mobile-report.html

As far as the malicious/fraudulent App reporting procedures are concerned, these differ from each app-store and the most common issue is that the reporting procedures are not clearly defined by the App-stores. The same goes for the removal/take-down procedures of a malicious App. The Google's Android Market for instance, does have an option to report a malicious/fraudulent App on its website, which can be used to request Android Market to review and remove an inappropriate App from its store but it could become a difficult task for an end user to find the "reporting option" on Android Market website.

Furthermore, there have been a number of instances whereby a malicious App was reported but it took a while until that App was taken down by the Android Market. Therefore, it can be stated that the process of identifying/reporting and removing/take-down a malicious App from the Android Market requires serious considerations and perhaps improvements as a result of those considerations.

http://www.developer.nokia.com/Distribut 1

In the case of Nokia's Ovi Store, there is a revocation process in place for malicious Apps. Revocation provides a final sanction to handle applications which may pose a threat to the Nokia users, its network or the Nokia's Ovi community as a whole to ensure those malicious Apps do not spread any further. Fig.3 outlines an App submission and signing process for Nokia's Symbian:

Based on the findings of this study, the next section of this paper will aim to propose common policies or best practices for Apps management in order to improve the security of mobile Apps ecosystem.

5.2 Mobile Applications Management Best Practices for Different App-Stores

1.While conducting a security review of the submitted

Apps, the App-Stores should consider a number of different known threats associated to the mobile Apps. These threats could include but are not limited to[8]:

- Activity monitoring and data retrieval
- Unauthorized premium rates dialing and premium rates SMS
- Mobile banking and mobile payments related frauds (since the mobile banking Apps are increasingly being uploaded and downloaded)
- Unauthorized network connectivity
- UI (Unique Identifier) impersonation
- Sensitive data leakage
- Unsafe sensitive data transmission

2. The App-Stores should increase the user awareness regarding malicious Apps and the consequences of downloading such Apps by introducing a number of different measures including an informative message, clearly visible on the App-Stores' websites. This should also be the case when it comes to the reporting procedures so that a user could follow simple steps in order to report malicious Apps to the relevant App-Store.

3. An introduction of a warning message for the Apps developers/authors could also be introduced, stating that a malicious App upload will lead to their accounts being deleted permanently and possibility of their details being passed onto the law enforcement agencies, in case of a criminal activity as a result of their App being downloaded by the users.

4. All App-Stores should look into placing a quality framework in the form of a series of standard security tests which could issue health certificates for different Apps in order to combat the growing threat of fraudulent Apps.

5. Apps Isolation - A strong security mechanism for ensuring mobile Apps access permissions could be introduced by the App-Stores, since it would not be sufficient to allow an App access to certain functions as the functionality needs to be further restricted by type, by time and conditions.

[8] *http://info.veracode.com/Whitepaper-2011-Mobile.html*

6. Banking and other Mobile Financial Services related Apps should only be allowed to be published by the banks and related financial institutions, ensuring their credentials are fully verified prior to submission of those Apps.

7. In order to improve the security, the App-Stores could carry out a pre-screening of the Apps and could also deploy a continuous monitoring procedure to ensure that the Apps which have been upgraded could also be screened / monitored.

8. A comprehensive vetting mechanism for Apps developers/authors could be deployed for all app-stores to ensure a successful tracking-down of the owner of the App in case of an App turning out to be a malicious App. For instance, all Apps publishers must have an account with the appropriate app store before they could submit an App to the app-store. Also, the app-stores should verify those publishers' accounts by emails, postal addresses, telephone numbers, and also through their credit card details.

   All App-Stores should consider deploying some sort of Apps and Apps developers/authors reviewing procedures including the review counters, which would automatically raise the alarm in case of a negative review. Perhaps an Apps auto-suspension procedure could also be deployed in case of reaching so many numbers of negative reviews (e.g. 5), until the App is investigated by the app-store security team. A serious consideration should also be given to monitor how those feedbacks were being submitted in order to ensure that the fraudsters could not abuse the review process.

10. An App review rating from one App-Store could also be used by other App-Stores through some sort of a shared reviewing mechanism implemented jointly by all app-stores. One of the most serious concerns here could be that the most users would rate Apps for their actual functionality and not for their security aspects. It would therefore be important to have two separate categories for the App reviewing mechanism, one for security and privacy issues, which could include an App asking for excessive privileges at install, and the other for the general functionality issues of an App, which could include information such as the App worked as it was supposed to, etc.

11. A continuous monitoring of the Apps developers/authors in a way that the prior Apps contribution should not be taken into account and each new submitted App is considered on its own basis.

12. The malicious Apps reporting procedures should be well defined and easily visible (highlighted) and accessible to the users of an App-Store. One of the options could be a simple reporting form in order to submit a complaint which should be dealt with and resolved, as quickly as possible. Another suggestion could be a live chat channel in place on all App-Stores so that a user who wishes to report a malicious App could do so, as and when needed.

13. A simple but effective and appropriate, “take-down” or “removal” procedure for all different App-Stores could be implemented. Perhaps a unified “shared” take-down mechanism could be employed throughout different App-Stores in order to ensure the malicious App publisher gets barred from all stores.

14. All App-Stores should have a kill-switch to remotely kill a malicious App upon reporting/discovery. A generic policy and procedure on “remote wipe” could also be implemented in case of a malicious App disaster ensuring that the users are fully aware of those procedures.

15. All App-Stores should consider priority vetting for updates of existing Apps in order to enable the Apps developers/publishers to patch up the vulnerabilities quickly and effectively.

16. A knowledge share mechanism between different App-Stores could be deployed in order to report and monitor a fraudulent developer’s/author’s activities.

17. A generic (for all App-Stores), mobile devices Apps downloading and user’s best practices could be developed in order to ensure an increased awareness for the end users.

6. CHAPTER III “nfactor authentication in ebanking environments”

6.1 Situation

The blooming of Internet technologies has revolutionized the way people transact with their financial institutions. E-banking users have considerably benefited from using e-Banking services without time and location constraints (‘whenever you want’ and ‘wherever you are’). Banking customers can also access their account balances and con- duct transactions not only via their personal computers but also via their smartphones and tablets. According to a recent survey by Ernst & Young [40], excellent online banking experience is one of the main attributes that banking customers value most, a fact that is indicative of the convenience, simplicity and speed those e-Banking offers.

Nowadays, it is estimated that one in four Internet users access banking sites globally [38]. Moreover, given the surge of mobile technology, the number of people that use their mo- bile device for e-Banking and e-financial services has drastically increased over the past years. According to Juniper [39], over 590 million mobile users have used their mobile devices for banking purposes.

On the other hand, the lack of proper security measures raises concerns about financial institutions’ security practices and acts as a barrier for the wider adoption of online and mobile banking services [20]. It is indicative the fact that security of online payments is the second most common concern of European Union’s citizens when using online banking ser- vices [41]. By taking advantage of the aforementioned poor security practices of online banking systems, cyber-criminals are able to conduct fraudulent online transactions that cause money loss to banks and their customers.

A recent study by European Commission reveals that 7% of EU Internet user- s say that they have been victims of credit card or banking fraud online [41]. In addition, the sensitive nature of financial and personal information used in online banking may raise privacy issues. If an attacker gets access to customers’ data, it can be regarded as a privacy

violation from the customer's perspective while the attacker can also use this information for further social engineering attacks.

Given the rapid evolution of cyber-crime, financial institutions should move forward. Since the implementation of e- identification and authentication methods (eIDAMs) is considered a key aspect in today's e-Banking security, they must be able to provide robust eIDAMs when accessing sensitive information and/or performing risky operations. Therefore, even though online banking services are slow to adopt current authentication methods [42], risk and cost-benefit analyses should be performed and subsequently the most suitable eIDAM(s) should be chosen and implemented.

6.2State of the Art

Cybercrime is increasingly targeting e-Banking systems ('criminals go where the money is') and at the same time getting more and more organized. Nowadays, cybercriminals have the capabilities to target new technologies and launch attacks of increased sophistication. Such examples are the two high impact attacks during 2012 that shocked the e-Banking sector. In the High Roller incident [43], 60 million euros were stolen while in the Eurograbber attack 36 million Euros were stolen and more than 30000 bank customers were targeted [44]. In both incidents, attackers were able to hijack two-factor authentication (e.g. mTAN) and commit fraud. As a result, e-Banking authentication continues to draw important attention to security professionals, and security re- searchers.

Authors in [45] discussed about a wide range of security challenges and issues identified in current e-Banking systems. Similar, author in [46] and [47] presented an extensive study on e-Banking security and introduced a formal definition of e-Banking threats and security models.
Security threats was also the focus of authors in [48], authors presented a study on the main security threats towards proposing the use of biometrics for strong authentication in e-Banking scenarios. In a similar form, authors in [49], study the security of e-Banking authentication, and discuss about biometrics adoption as a second or third authentication factor.

A taxonomy of attacks to e-Banking authentication was introduced by authors of [50], additionally a challenge/response authentication solution has been proposed. An assessment of authentication methods for e-Banking was presented by authors in [51] however; the presented study was focused only on the PC-based scenarios.

Although interesting approaches, the evolution and sophistication of Internet and mobile attacks require a continuous study of emerging threats, issues and challenges especially design to attack e-Banking scenarios. Similar to our study, authors in [52] analyzed the eIDAMs used by major banks in English speaking countries; nonetheless, their research was limited to data collected by mere observation (publicly available), contrary to our approach, which, analyzes these factors, taking into account current e-Banking systems with information provided by the security professionals of major European banks.

Finally, our study provides an eIDAMs assessment and a set of recommendations taking into account the security professional's perspective.

6.3e-Banking Threats Categorization

This section identifies the most relevant security threats in e-Banking and explains how those threats are related to the authentication process. The categorization of threats is based on the asset that is being attacked (Figure 1); this approach was firstly introduced by authors in [46].

A. Threats against end-users

End-users are often seen as the weakest link when ensuring security. In particular, identify theft can be achieved by i) physical observation, ii) social engineering or iii) phishing campaigns.

1) Physical observation

direct observation techniques like shoulder surfing can result in identity theft, user's credentials can be captured or even stolen.

2) Social engineering

through non-technical means, users' credentials can be com- promised, just by having knowledge of personal user details; this can result in user impersonation, like claimed identity during registration, phone calls, etc.

3) Phishing

End-users are often victims of traditional phishing email at- tacks, also highly targeted spear-phishing or heterogeneous phishing campaigns (e.g. phishing through online social net- works). This attack allows the attacker to collect users' credentials, often by leading end-users to phishing sites.

B. Threats against end-users' devices

Threats to end-users' devices are considered those targeting various devices like PC, mobile devices, tokens, etc.

1) Device theft

End-users may be victims of theft of their token or mobile device, or even a piece of paper with their passwords writ- ten down; combined with other information or attacks, this attack may result in credentials theft.

2) Tampering

Of token or device, PIN brute force or side channel attack- s, hardware key-loggers or even replication are some attack examples of this threat.

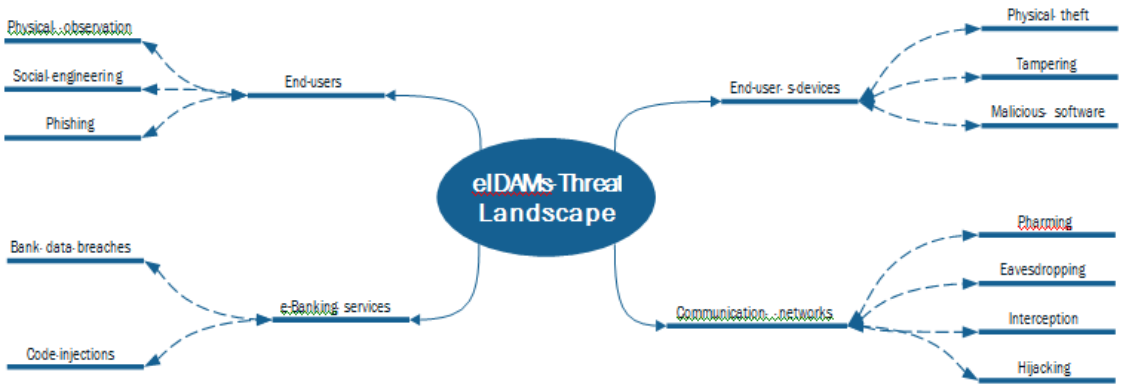


Figure. 1: Threats against e-Banking systems

3) Malicious software

Malicious software authors are increasingly targeting electronic banking systems. The embedding of malicious code aimed at stealing credentials and other sensitive information may take place through different methods. Examples of those methods are keyloggers aimed at recording users' credentials and mobile banking Trojans, which, in turn are able to hijack two-factor authentication.

C. Threats against communication networks

Threats targeting the communication channel between end- users and the remote banking server.

1) Pharming

Pharming attacks target the network infrastructure (specifically routers and DNS servers) and redirect end-users to illegitimate websites, i.e. users are redirected to a malicious URL while having entered the correct one in the address bar. Credential theft if possible in a similar form as in phishing attacks.

2) Eavesdropping, interception and hijacking

This threat category includes network-based and server- based man-in-the-middle attacks. e-Banking credentials, transaction data, OTPs and other sensitive information is captured through i) erroneously trusting in fake SSL certificates;

ii) passive traffic monitoring (sniffing); iii) replay attacks; and iv) active man-in-the-middle attacks.

D. Threats against e-Banking services

These threats target the electronic banking infrastructure that hosts the web banking service and the respective data stored.

1) Code injections

Web code injections against banking servers allow attacker- s to exploit vulnerabilities of the web banking services and applications; these attacks are performed through different web exploitation techniques, such as SQL injections, cross- site scripting, cross-site request forgery, and redirection to a malicious URL. These attacks allow information collection such as users' credentials.

2) Bank data breach

Internal or external attacks to banks' databases allow data breaches of sensitive information (e.g. end users' credentials, account information, social security numbers etc.).

6.4e-Banking Authentication

Authentication is the process that allows an entity to establish the identity of another entity. Current eIDAMs are based on:
i) Something that you know, ii) something that you have, and
iii) Something that you are. When two or more of those elements are combined the process is known as strong or multi- factor authentication. Additionally, authentication based on where the user is or what the user does is known as context- based authentication and it is commonly used to complement strong authentication methods. During the initial stage of this research, we have selected the most common and relevant eIDAMs for e-Banking. The categorization has been done according to the type of element and credentials. A short description of each eIDAM is

introduced in the following subsections.

iv)

A. Knowledge-based authentication

Authentication approaches based on what you know usual- ly consist of a combination of a valid and unique identifier (username) and a secret pass-phrase (PIN or password). There exist two important security enhancements to this approach namely i) virtual keyboard and ii) partial password. The virtual keyboard is a software-based keyboard displayed on the screen aimed at preventing keyloggers; while in the partial password approach, users are asked to enter only some of the digits or characters from the PIN or password, usually partial password is implemented to request different position- s for each session. The combination of both (virtual keyboard and partial password) provides a higher level of protection against keyloggers.

B. *Possession-based authentication*

The two widely adopted authentication approaches based on what you have are i) One Time Password and ii) electronic Signature.

1) *One Time Password*

One Time Password (OTP) consists in generating a different and essentially random password namely OTP, which, is valid for only one session or transaction. OTP implementations are basically based on either event-based, time-based or challenge-based algorithms. OTP-based authentication can be delivered in several forms providing different benefits in terms of security, usability and costs, those forms are de-scribed below.

- Static OTP: physical documents containing a list of codes (OTPs) to be used in each transaction like coordinates cards or TAN code lists.

- Dynamic OTP: the OTP is delivered dynamically activated by the transaction, examples of these approaches are i) SMS-based OTP, the OTP is sent via an SMS message to the registered phone number; ii) hardware token consisting of a physical device generating time or

challenge based OTPs; iii) software token consisting of a software application installed in users' device generating time or challenge based OTPs, a novel enhancement of this approach is through the use of QR-codes.

2) *electronic Signature*

In electronic signature (i.e. Public Key Infrastructure - PKI) approaches, end-users are authenticated through e- Signatures. The level of security and usability of this approach is highly influenced by where the user's private key has been stored.

- Computer stored key: users' private key is stored in their own PC's hard disk.

- Mobile phone: users' private key is stored in the mo- bile device, either in the SIM card or any other secure element present in the device.

- Memory card: users' private key is stored in a USB or a SD memory card namely removable memory storage.

- Crypto card or smart-card: users' private key is stored in the chip of their e-Identity card or specific smart card with cryptographic embedded functions

C. *Inherence-based authentication*

Authentication based on what you are is based on biometric methods, traditionally distinguished by the type of characteristics that they evaluate like physiological attributes or behavioral singularities. These section overviews the leading biometric techniques commonly used to verify the identity individuals.

1) *Physiological biometrics*

Physiological biometrics consists of measurements taken from data obtained as part of the human body, leading techniques in this category are introduced below.

- Hand geometry: this process extracts features from the hand such as shape, appearance, textures, length and perimeters of fingers.

- Fingerprint: identifies the lines convergence points.

- Facial recognition: a process that captures a sequence of images, and extracts features from the images ensemble to build the biometric template. .

- Iris recognition: identifies the location, shape and size of the random patterns in the external iris of the eye.

- Retinal: the retinal systems scan the blood vessels in the back of the eye, they are highly accurate and require low storage.

2) Behavioral biometrics

Behavioral biometrics consist of measurements taken from the user's actions, many of them are indirectly measured from the human body such as voice. The leading behavioral techniques are introduced below.

- Keystroke dynamics: is the process of authenticating [56] an individual based on the verification of user's typing pattern. This process measures and compares the series of user specific timing events also known as "typing signature". Traditional approaches are based on data obtained from conventional keyboards, and determines specific metrics, such as, the dwell time and the flight time that are ultimately represented as math series.

- Voice recognition: makes a power and spectral analysis of several

samples of speech, building a statistical pattern from all of them.

- Handwritten signature: signature verification systems use a digitalized version of the signature. Modern sensors are however able to additionally measure on each point, pen positions, pressure and inclination all in a three-dimensional way [55].

D. *Context-based authentication*

Also referred as continuous authentication, this authentication method takes advantage of contextual information and it is recommended as a complement to other strong authentication methods. Context-based authentication improves the level of confidence of service providers regarding the identity of a user, independently of the provided user credentials. In most cases it is implemented at the server side and transparent to the user. The most relevant forms of continuous authentication in e-Banking, determine users' authenticity by analyzing users' behavior, users' scenario and transactions.

1) *Behaviour authentication*

The behavioral authentication process constantly analyzes contextual information that characterizes the user's interaction with the e-Banking system. Implemented approaches strongly depend on the type of information available to the e-Banking system. The following validations are suggested:

- Individual behavior pattern: users' behavior is analyzed in all sessions, creating individual patterns. An individual user profile (history) is stored in the system and compared (current and historical sessions).
- Multiple behavior patterns: users' behavior is analyzed during active sessions. Validation is done by comparing user's pattern against several "standard" profiles (e.g. company profile, end user profile, investor profile). If users' behaviors do not match with the profile associated to them, it may cause restrictions to certain operations.

2) *User scenario authentication*

The scenario authentication correlates unique and contextual (network

session) information from a registered user device. Detected patterns are compared at every session. In particular device authentication is used to reinforce user's identity. When different device data is detected, a scoring is computed and a warning is usually triggered by the continuous authentication modules, possibly resulting in communications being blocked for a particular device.

- Platform and device identification: fingerprinting of de- vices and platform information are evaluated at every session. The parameters considered include: i) device unique characteristics (e.g. MAC address); ii) Hardware characteristics (i.e. CPU model, clock speed, memory latency, memory size); iii) user key stored in the device;
  v) Platform characteristics (i.e. OS and browser versions).
- Network session identification: contextual information related to the network characteristics are analyzed. In particular validations regarding the status of an IP ad- dresses are performed such as white-listed IP addresses, anonymous proxies or blacklisted IP addresses.
- Geolocation: IP address geolocation, or other geographical data allows the system to evaluate the origin of a transaction and the associated risk. Various anomalies can be detected in particular in the cases of i) IP ad- dress' belonging to a high risk country classification;
  ii) Inconsistencies in the geographical distance between operations, and iii) short times between sessions from a single source IP address used to access several customer accounts.

3) *Transaction authentication*

Continuous evaluations based on users' banking account behavior are commonly performed by financial services in or- der to authenticate transactions. Tracking users' overall e- Banking operations and transaction history when interacting with the e-Banking service allows prevention and detection

of fraudulent activity by evaluating deviations from the expected behavior. This authentication category includes:

- e-Banking sessions: each user interacts with e-Banking services and makes transactions in a unique manner. Parameters such as the time of the day to perform operations; the time between operations

within a session, and the frequency of accessing the service within a time window are evaluated.

- Destination account: based on users' transaction history and transactions to unknown or blacklisted account numbers, transaction may be rejected.
- Amount in operation(s): based on users' profile, the amount of a single operation or of operations within a period of time is compared and evaluated. Operations not compliant with accepted patterns might require additional user interaction.

6.5 eIDAMs Evaluation and Selection Criteria

This section first proposes a classification of operation types in e-Banking systems followed by a categorization of targeted e-Banking users. Both needed to identify the selection factors required to simplify the evaluation of eIDAMs' suit- ability.

A. *Operation types*

e-Banking applications allow users to perform various types of operations. This section proposes a classification of those operations according to four levels of risk associated to them. This classification partly follows the Data Protection Authority's (DPA) recommendations for data breach severity evaluation.

- Operation type 1: general customer information (not financial), allows only read-access and does not provide any substantial insight of customer's financial information.
- Operation type 2: account information (read only), al- lows only read-access to personal and financial data. An attacker having knowledge of this information can commit other means of fraud.
- Operation type 3: safe transfers and e-Payments to registered or trusted accounts; it allows payments to trusted destinations, relatively simple to undo in case of fake or erroneous requests (e.g. Utility payments).
- Operation type 4: risky transfers to untrusted or unregistered accounts

(e.g. transfers to any destination, investment instruments, e-Merchant payments, management of personal data).

B. *User segments*

Several financial institutions target different types of users, this is generally known a user segments. Having a categorization of them simplifies the suitability assessment of available eIDAMs, especially in terms of costs. This section introduces a generic categorization of user segments. Note that not all financial institutions might include all of the pro- posed user segments; there exist those institutions aimed at only one of these categories.

- Retail: personal banking or consumer banking, aimed at regular customers making low amount transactions.
- Private: personal banking with special needs, i.e. customers making large amount transactions.
- Corporate or Business: aimed at companies/enterprises.
- Investor: aimed at users managing investment tools.

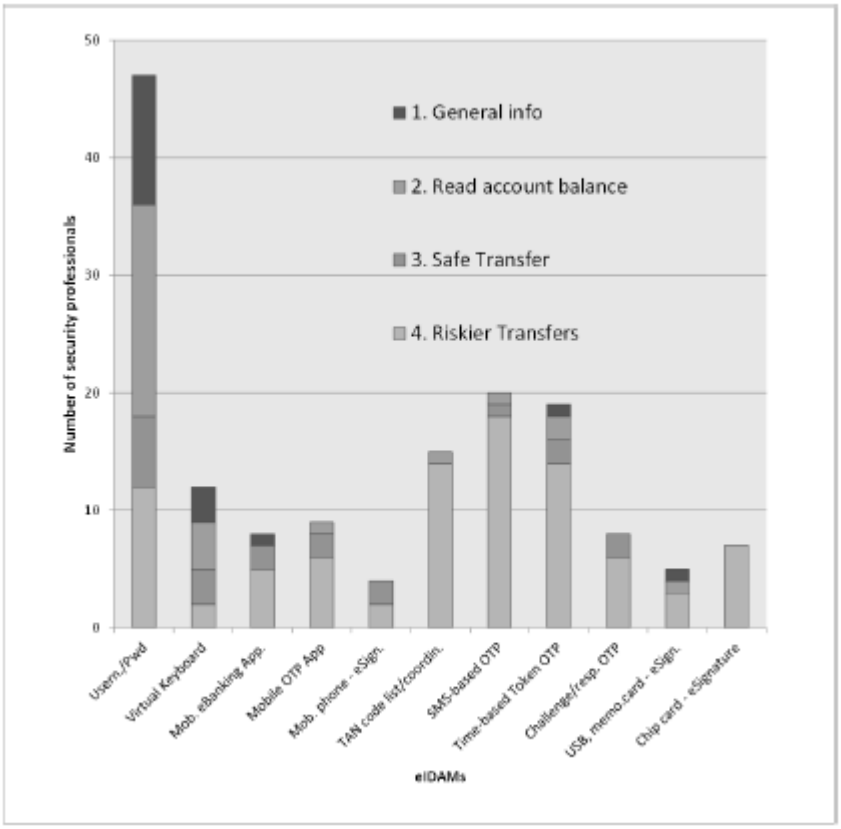
C. *eIDAMs selection and evaluation factors*

Selection factors used in real-life implementations, i.e. considered by security professionals, managers and decision makers in the e-Banking sector, are briefly introduced in this section.

- Security: the strength of eIDAMs in terms of covered risk and efficiency to resist potential attacks in relation to the represented risk and sophistication.
- Usability: eIDAM's quality of being user-friendly and closer to user needs (ease of use).
- Implementation cost: the cost impact of the eIDAM's implementation and maintenance effort.
- Implementation difficulty: in terms of technical requirements to be fully deployed and functional.
- Adequacy: the quality of being able to meet the needs and

expectations of a particular user segment (Section V-B).

- Average loss and reduction: by means of financial loss per involved customer and loss reduction after implementing a specific eIDAM.
- Security incidents, frequency and reduction: yearly frequency and reduction of security incidents including number successful attacks per year, regardless the number of customers involved on each of them.



Additional factors that are specific to biometric-based implementations have also been considered in the evaluation of eIDAMs and are introduced below.

- Biometric permanence: the condition that biometric data should not change over time.
- Biometric reliability and accuracy: a biometric system is only considered reliable if the performance rates are acceptable. Two conventional metrics used to evaluate biometrics' performance are i) false acceptance rates and ii) false rejection rates.

6.6 eIDAMs Evaluation Analysis

The categorization of threats, operations, user segments and eIDAMS described in Section III, Section IV and Section V respectively, have been proposed to support a survey ad- dressed to 160 security professionals in the e-financial sector in Europe. The objective of the

survey was to identify the eIDAMs that are currently implemented in e-Banking systems. The evaluation of implemented eIDAMS according to the se- lection factors introduced Section V has also been carried out as part of the same survey. The survey has been distributed and conducted by the European Network Information Security Agency with the contribution of the APWG.EU, CaixaBank, Merchant Risk Council, SecuRePay, FI-ISAC, ECB, EPC, and the FSUG.

A. Current practices

Several eIDAMs have been introduced in Section IV; how- ever, only those represented in Figure 2 have been widely adopted. Figure 2 shows the most implemented eIDAMs associated to the operation types (Section V-A); virtual key- board is only a security enhancement of username/password approaches, but it was important to highlight the number of security professionals implementing such enhancement. As it can be observed most banks implement knowledge-based authentication, but only for non-risk operations (types 1 and 2), nevertheless, a few of them still implement weak eIDAMs for risky operations. In turn, the possession-based authentication methods are used by most of them to authorize risky operations (type 3 and 4). Note that

Figure. 2: eIDAMs implementation by European banks for each operation type

Surprisingly, inherence- based authentication (i.e. biometrics) has not been implemented yet for e-Banking systems in Europe, Section VIII will provide a further discussion regarding the adoption of biometrics in e-Banking. Contrary to biometrics, security professionals introduced the mobile application-based authentication, which is not an authentication method itself but a way to deliver it and a current trend in e-Banking systems. It is also important to notice that in most cases security professionals that contributed by answering the survey did not make any distinction between operation types 1 and 2, in the same way it is difficult to make distinction between types 3 and 4.

B. Applicability of identified threats to eIDAMs

Table 1 presents the applicability of the identified threats (Section III) in the different eIDAMs. It has been assessed taking into account common practices in e-Banking, recent literature [46], [51], and real world attack scenarios.

Table 1: eIDAMs Threats Applicability

Authentication mechanisms (eIDAMs)	Threats									
	A1	A2	A3	B1	B2	B3	C1	C2	D1	D2
Username/password	5	5	5	3	3	5	5	5	5	5
TAN code list	4	3	5	5	5	5	5	5	5	5
SMS-based OTP	1	1	4	3	2	5	4	5	4	5
Time-based h/w OTP	1	1	3	5	3	5	3	5	4	5
Chal. Resp. h/w OTP	1	1	3	4	2	4	3	4	3	5
Mobile OTP App	1	2	3	3	3	5	3	5	3	5
Token e-Signature	1	1	1	4	2	4	1	3	2	3
Chip card e-Signature	1	1	1	3	1	4	1	3	2	3
Mobile phone e-Signature	1	1	1	3	2	4	1	3	2	3

Threats are described in Section III above.
Applicability ranges from 1: Not Applicable to 5: Highly Applicable

As shown in Table I, most of the identified threats are highly applicable to the username/password eIDAM. Moreover, only malicious code (i.e. threat B3) and bank data breach (i.e. threat D3) are applicable to all eIDAMs. Physical attacks against devices or tokens (i.e. threat B1) are mostly applicable to the OTP and e-Signature eIDAM categories. Tempering of devices, (i.e. threat B2) applicability depends on the robustness of the device. Finally, phishing (i.e. threat A3) apart from being applicable to Username/password and TAN code list is also applicable to the OTP eIDAM category.

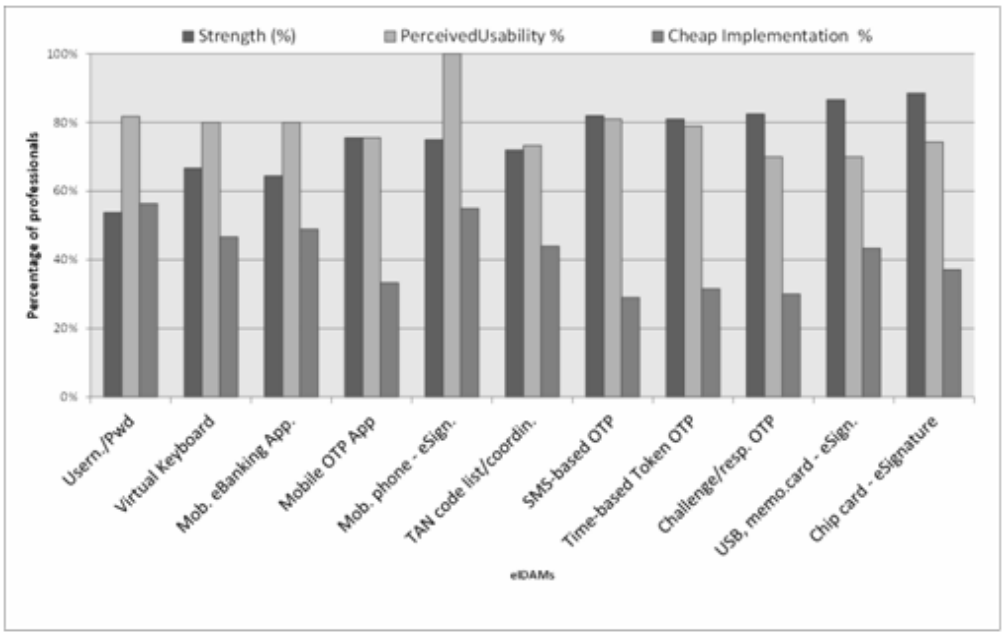


Figure 3: eIDAMs perceived strength, usability and cost

C. *Perceived strength, usability and cost*

Through the responses of the security professionals, we have identified quite clearly three groups of eIDAMs

according to its strength, which suggests its suitability for the operation types (Section V-A). Figure 3 also confirms the common practice in the financial sector of giving priority to usability over other criteria. The usability restriction for implementing stronger eIDAMs should be avoided through adequate training and awareness of customers, and also improving the adoption strategy and the "user experience" through adequate developments.

D. *Adequacy of eIDAMs' selection criteria*

Figure 4 and Figure 5 show the relevance of the different selection criteria for implementing eIDAMs for the medium and high strength groups of operations respectively.

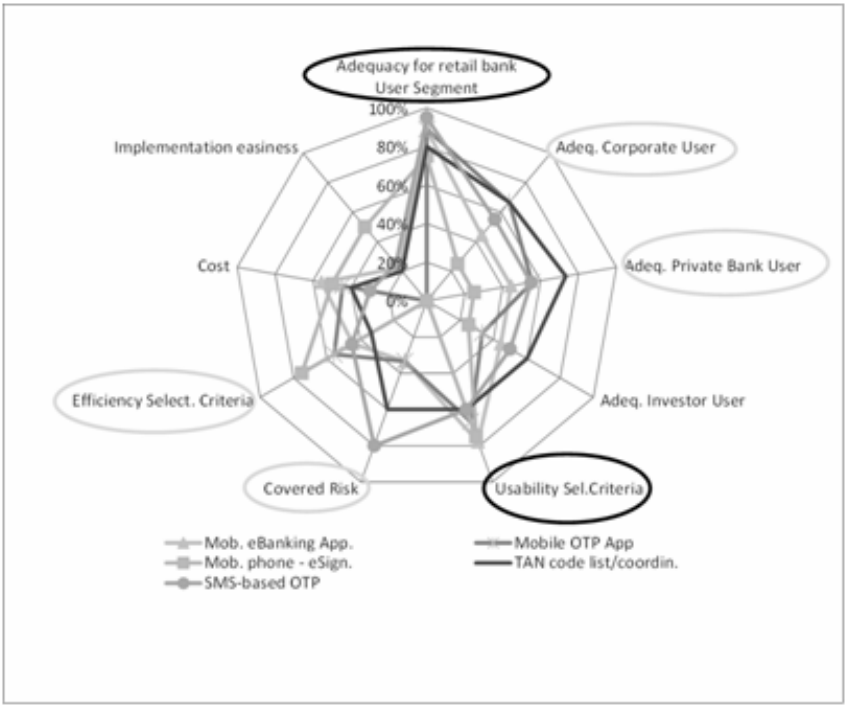


Figure 4: eIDAMs adequacy of selection criteria(medium strength)

Circled in dark gray, are the most relevant criteria for each strength group, and in light gray the less characteristic ones. All eIDAMs involving the use of hardware tokens are more suitable for corporate, private and investor customer segments (Section V-B). Distribution costs and usability (an- other device to be carried) drawbacks are less relevant for those customer segments, because the number of customers is significantly lower (<1% of retail users), and they are more conscious of their transactions risk, thus, accepting stronger eIDAMs to be applied. New formats of token devices are now emerging and are able to improve usability (e.g. token credit

card format). However, the associated costs are even higher, narrowing the profiles of the customers to which they could be offered.

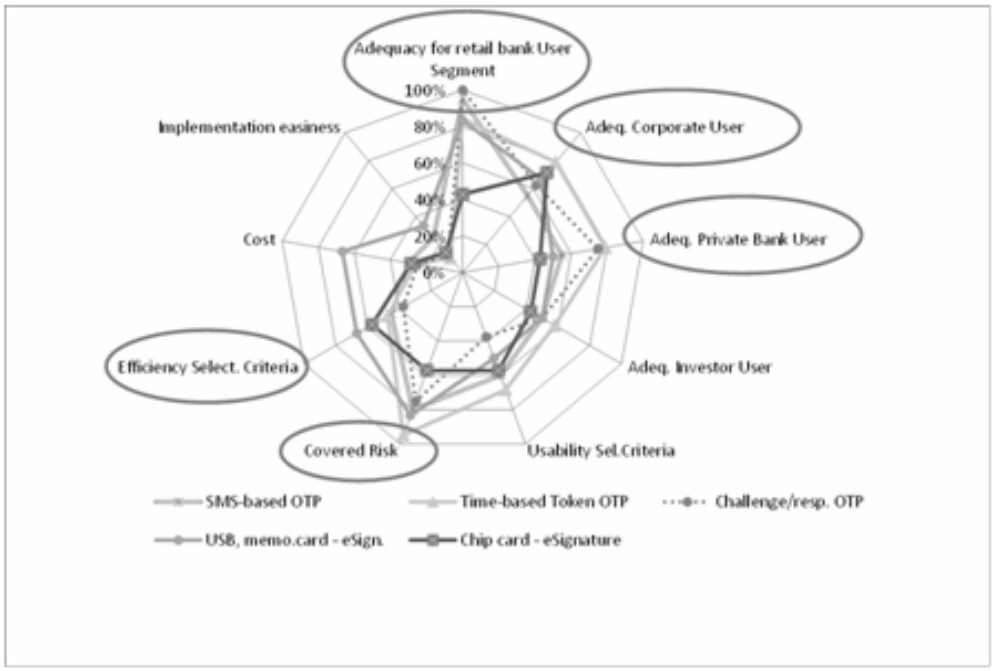


Figure. 5: eIDAMs adequacy of selection criteria (highest strength)

E. eIDAMs risk reduction benefits

Figure 6 shows the summary of the reported "loss", "loss reduction" per user and incident, and "number of incidents" associated to each eIDAM. The most frequent eIDAMs have also the higher risk, because they attract the interest of criminals to research and develop more sophisticated attack pat- terns and tools. One exception is the Challenge OTP, from which the reported loss is very high, reflecting that it is used only for the customers that perform the most risky operations, but the number of incidents is one of the smallest, and so the total loss for threats to this method keeps small. The opposite case is the Token (USB, memory card) e-Signature, with relatively large number of incidents and very low loss for each, because of the difficulty to use the token by other person rather than the owner, and because this method is also used for low risk operations. In general, the results of the survey show that the lower number of incidents corresponds to the less implemented methods; mostly those based on mobile handset token OTP or e-Signature. In summary, each financial organization has to perform the cost benefit analysis, taking into consideration those parameters for their own environment, because their number of users and actual costs will influence in the total savings due to

each eIDAM.

6.7 eIDAMs Continuous Authentication Evaluation Analysis

The second phase of the survey analysis is aimed at identifying the most common continuous authentication methods currently implemented in e-Banking systems in Europe. Based on the security professionals' responses, an analysis of different factors are discussed including perceived strength; operation types associated to different continuous authentication methods; their adequacy to user segments (Section V-B and selection criteria (Section V) priorities that are given to each method.

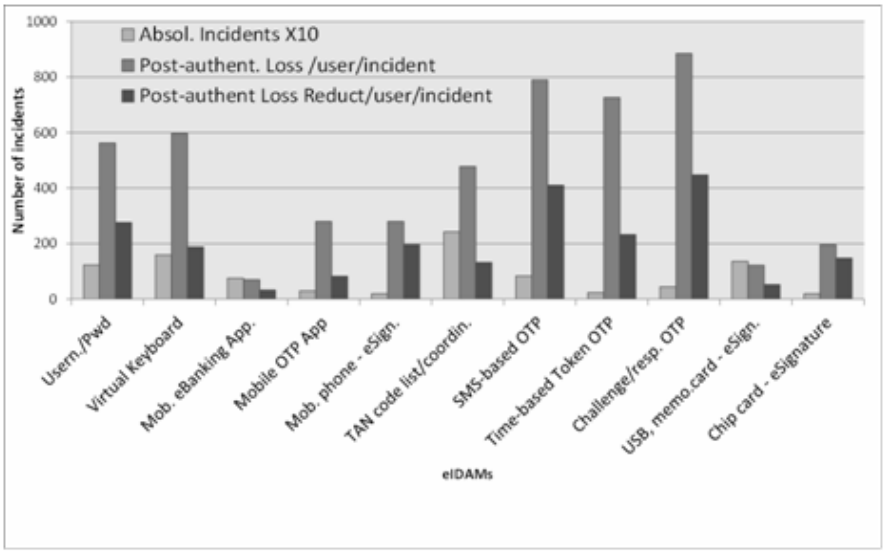


Figure. 6: eIDAMs reported incidents, loss and loss reduc- tion

A. *Current practices*

Figure 7 presents the most frequently implemented continuous authentication methods as it have been categorized in Section IV-D. As shown in Figure 7, transaction authentication systems are widely adopted in e-Banking systems.

While network and geo-location parameters are also taken into account, it can be observed that user profiling and plat- form

identification have low implementation rates, the reason might be that user profiling and platform identification are the emerging methods. In this regard, it is expected that due to the rapid evolvement of mobile computing together with the fact that current devices allow us to gather more useful information; the implementation rates for these methods will severely increase in the coming years.

B. *Perceived strength*

Results show that security professionals consider two group- s of continuous authenticati on methods according to their strength, which suggests their suitability for the previously introduced operation types. Figure 8 shows that there are two well defined groups of continuous eIDAMs, based on their implementation for safe transfers (left and right of the vertical line). It is also worth to mention that, the two eIDAMs with lower perceived strength ("blacklisted accounts" and "high risk countries") are some of the most implemented, probably due to their adequacy to all user segments (see Figure 9).

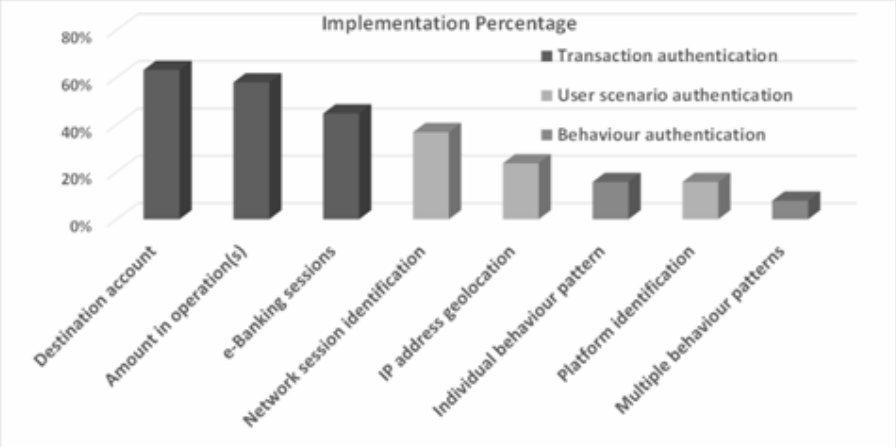


Figure. 7: Implementation rates of continuous authentication methods

Figure. 8: Perceived strength and associated operation types

c. Continuous authentication: adequacy of selection criteria

Figure 9 and Figure 10 show the relevance of the different selection criteria when implementing continuous authentication mechanisms in both lower and higher strength groups. The most popular selection criteria taken into consideration to implement methods from the lower strength group (Figure9) are the following: i) efficiency and adequacy to the corporate user segment for methods evaluating the total amount in operations; ii) implementation easiness for anonymous proxy evaluation; iii) low costs and usability for evaluating time between sessions; iv) covered risk by methods evaluating the geographical distance between IPs; v) high risk country classification is adequate to retail user segments; and, vi) blacklisted bank accounts are suitable to private and investor user segments. The

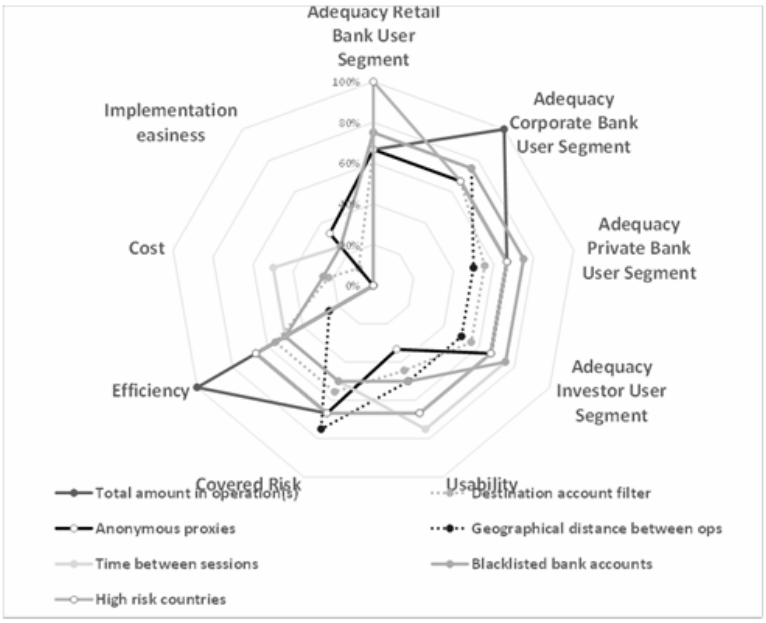


Figure. 9: Selection criteria priorities used in lower strength continuous authentication methods

most popular selection criteria taken in- to consideration when implementing continuous authentication of the higher strength group (Figure 10) is explained be- low. Implementation easiness is the selection criteria used when implementing time between operations (one session) method; adequacy to retail user segment is the main reason for implementing white listed IPs, while adequacy to corporate user segment and covered risk are the influencing factor for implementing blacklisted IPs and platform identification. Time between sessions evaluating methods selected due to their adequacy to private and investor user segments, and finally the amount filter evaluation method is chosen because of its low costs and usability.

6.8 Biometrics Adoption in e-Banking

Results of the study on eIDAMs adoption for e-Banking in Europe have shown that despite the potential security attack- s, the most implemented mechanisms continue to be those based on what you know (e.g. TAN code list) and what you have (e.g. OTP-based hardware token). The main weakness of those mechanism s is their inability to provide non-repudiation , which, is especially needed in the e-Banking ecosystem. The non-repudiation feature cannot be provided by any method that can be either replicable or transferable. Therefore, to cope with this limitation, biometric solution s have been strongly suggested in the past.

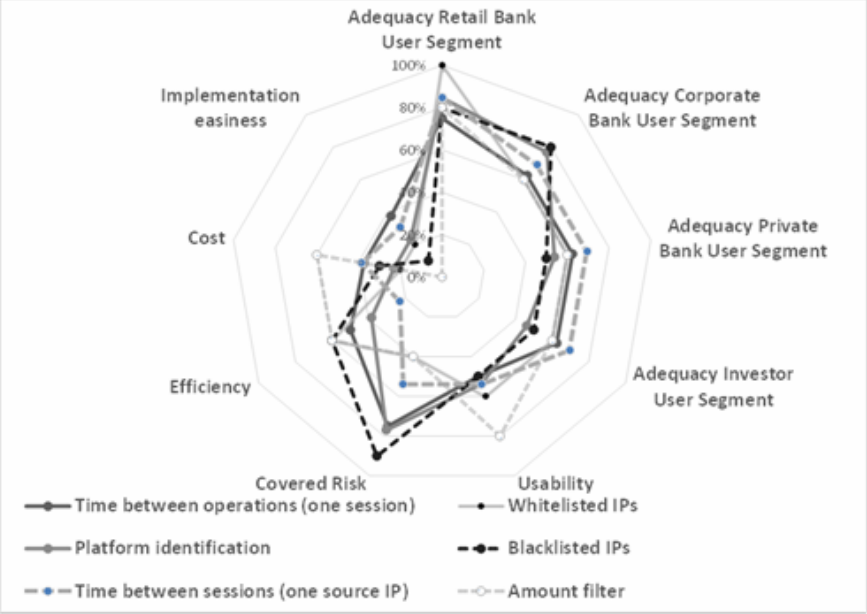


Figure 10: Selection criteria priorities used in higher strength continuous authentication methods

Nevertheless, results of the study have shown that very few banks in Europe implement biometrics as authentication method for e- Banking. The rationale behind this phenomenon is discussed next.

A. *Challenges on biometrics adoption*

1) *Security*

False Acceptance Rates (FARs) that are inherent to biometric technologies, must be also properly evaluated. In addition, the proper implementation of biometric-based systems in terms of security is difficult to achieve. Thus, potential at- tacks to the overall system implementation must be carefully addressed.

2) *Usability*

False Rejection Rates (FRRs) can hinder the suitability of different biometric technologies, user are reluctant to adopt solutions that are not able to authenticate them at the first try Usability is therefore a main concern and it should be carefully evaluated when selecting appropriate biometric technology.

Table 2: Biometrics Suitability Assessment

	Security	Accuracy	Permanence	Usability	Implementation difficulty	Costs	Adequacy
Fingerprint	H	H	M	H	L	M	C,P,I
Facial	M	M	M	H	M	L	R,C,P,I
Iris	H	H	H	L	H	H	P,I
Hand geometry	H	M	M	M	M	H	C,P,I
Retina	H	H	H	L	H	H	P,I
Voice	M	M	L	H	L	L	R,C,P,I
Keystroke Dynamics	L	L	L	H	L	L	R,C,P,I
Handwritten Signature	M-	M-	L	M	M	H	C,P,I

Legend: H=High, M=Medium, L=Low

R=Retail, C=Corporate, P=Private, I=Investor customers' profile

1) Data leakage

High associated risks, mainly due to the potential attacks to a centralized data base storing biometric data, are non- acceptable by the e-Banking sector. Even if the information is encrypted or hashed, due to the sensitive nature of biometric information, data is compromised forever and cannot be changed (e.g. fingerprint, iris, etc.); therefore, it results in both, high risk and great responsibility to be accepted.

2) Law enforcement

Existing legal issues regarding the storage, use and process of personal information are of utmost importance. In Europe, a explicit authorization from customers is required, which is a difficult task to be achieved.

3) Privacy concerns

Strongly related to Law Enforcement, people are unwilling or do not feel comfortable with granting permission on the storage of their biometric

information.

A. *Suitability assessment of biometric techniques*

This section presents an evaluation of the suitability of the leading biometric techniques for e-Banking systems; the e- valuation has been done according the selection criteria introduced in Section V and summarized in Table 2.

Biometric-based authentication has reached certain level of maturity and is considered the future of authentication. It can increase the overall security of an e-Banking system, it provides non-repudiation thanks to its difficulty to be replicated or forged. It is assumed that it can neither be shared, nor distributed, forgotten or lost. Its main advantage is that at least theoretically the authenticating individual must be present. However, current solutions may only be considered as a second factor or part of a multi-factor authentication sys- tem, either because of usability or associated costs issues as explained below.

1) *Fingerprint*

This technique is one of the most traditional forms of biometry; it provides a high level of recognition accuracy and it is relatively easy to use. It does not require complex user- system interaction; therefore, advantages of this method are its usability in terms of performance and ease of use and high user acceptance [53], ultimately resulting in a wider adoption of fingerprinting in physical and digital access control systems. However, despite the benefits, the adoption of finger- printing in e-Banking is affected by other usability factor i.e. users requiring an extra device (fingerprint scanner) and the overall system implementation costs; although, fingerprint s- canners are now considered low cost and small sized, delivering specific hardware (scanner) to millions of e-Banking customers, may result in a large economic impact, surpassing even the losses of a single entity. Fingerprinting techniques could work better to the corporate, private and investor user segments (i.e. a smaller percentage of customers performing high risk transactions). Nevertheless, technology trends have envisioned the integration of biometric scanners into modern mobile devices (e.g. iPhone 5 and Samsung Galaxy S5 are

now embedding fingerprint scanners), promising a wider adoption of biometric authentication at relatively low cost for end service providers.

2) Facial recognition

Facial recognition has become very popular, it benefits from high user acceptance. Biometric data extraction can be easily achieved without the need of specific sensors, therefore its implementation can be low cost when taking advantage of modern technologies, such as mobile devices equipped with embedded cameras. In principle, this technique is suitable for all user segments; nonetheless, it is of common knowledge that illumination conditions may affect the overall performance, resulting on limited usability.

3) Iris recognition

Iris pattern matching is a highly accurate technique. The in- formation richness of the iris makes it one of the most efficient forms of biometrics, providing a high level of security. Similar to facial recognition, it requires a camera to ac- quire the biometric data (sample taken from iris). Iris recognition systems are relatively easy to operate; however, different types of noise might impact their performance. Captured samples in non-controlled environments might suffer from poor image quality (e.g. blur) and illumination reflection-Effective systems, such as kiosks, are quite expensive and not suitable for user segments of e-Banking scenarios. Alternative to kiosk-based solutions, rely on the use of desktop or mobile cameras, resulting in cheaper costs but increased difficulty and consequently limited in usability. The impact in usability is mostly due to variable user distance to the camera and illumination conditions.

4) Hand geometry

Hand geometry is a very attractive biometric technique; it provides a user-friendly interface and requires low data tem- plate storage. Hand geometry is in general terms suitable for verification, hand data can be exploited in many forms and has high user acceptance. Its suitability for access control is further reinforced by the fact that, hand geometry scanners have relatively low costs. However, for e-Banking where

millions of users will require a specific device (scanner), the overall costs are significantly increased, limiting the suitability to corporate, private and investor user segments.

5) Retinal recognition

Despite their outstanding performance in terms of security, accuracy and permanence [54], they are not yet widely deployed. The rationale behind this, is due to its limited usability and high costs. Users do not feel comfortable mainly due to the light that shines to illuminate the iris vessels; moreover, the required high resolution scanners are quite expensive resulting in unaffordable costs for the financial sector.

6) Voice recognition

Voice recognition systems are able to provide authentication in a unique and non-intrusive form. They provide high acceptance rate because of the high usability [55]. The hard- ware requirements of voice authentication represent no bigger costs, since microphones are use to capture the biometric data and they are readily available. Moreover, the nature of voice recognition makes it suitable for all user segments. On the other hand, voice authentication implementations need to take special care of the required storage and performance, which, tend to be high. Furthermore, voice authentication cannot be considered as a unique factor of authentication; similar to other behavioral-based methods, during the training phase this method present low accuracy. Additionally voice recognition algorithms must be tolerant to noise and should not be influenced by variations of the voice produced by sore throat or cold.

7) Keystroke dynamics

Keystroke dynamics is considered suitable for all user segments and it represents almost no costs, since, no especial devices are required. Usability and acceptability are considered high because in most cases it can be performed transparently to the user. The main drawbacks of this technique are its low accuracy and low security level during the training phase; therefore, it can not be used neither as first factor, nor as second factor authentication. Nevertheless, it is suitable for any continuous

authentication system.

8) Handwritten signature

Handwritten signature is widely deployed and well accepted, although, it is also target of many security and forgery attacks. One of its main issues is permanence, due to the signature time variability evolving over time, the resulting signature is significantly modified. As a result, its associated security and accuracy levels are considered to be very low. Note that similar to other techniques, handwritten signature requires specific hardware, limiting its suitability to corporate, private and investor user segments.

6.9 Challenges and Recommendations

This section summarizes the main challenges identified in our research and the proposed recommendations to overcome those challenges.

A. Challenge: Promote adequacy of eIDAMs to context

1) Recommendation

the strength of the implemented eIDAMs has to be proportional to the risk associated to the operations they grant access.

2) Recommendation

medium- and high-risk operations require “Strong customer authentication” i.e. two or multi-factor authentication. In particular, it is recommended to avoid the sole use of the “something users know” approach.

3) Recommendation

medium- and high-risk operations require a strategy that includes the implementation of at least two eIDAMs either using different channels of communication or two different devices.

4) *Recommendation*

Implementation of continuous authentication eIDAMs may provide additional factors, improving security without additional impact on usability.

B. *Challenge: Improve awareness and behavior of customers and professionals*

1) *Recommendation*

continuous training of professionals and advising customer- s are needed activities in order to improve their perception of actual risks; keeping in mind the latest threats and attack patterns adopted by criminals and associated to eIDAMs and e-Banking transactions.

2) *Recommendation*

professionals of financial institutions have to guide their customers in order to i) overcome the perceived lack of usability (sometimes in appearance) of stronger eIDAMs required by e-Banking systems; and ii) increase the comfort and the awareness of customers regarding stronger eIDAMs that en- able better protection to their assets.

3) *Recommendation*

4) A general statement about implementation of continuous authentication eIDAMs should be announced to customers, and it should be done according to Personal Data Protection Directive. *Recommendation*

technology providers must guarantee secure e-Banking application development; taking into consideration current and future threats to Operating Systems (e.g. mobile attack vectors) and performing data security analysis (persistence and access control).

5) *Recommendation*

Distribution of e-Banking applications has to be done through trusted channels and reputable sites that are able to guarantee that applications have been tested for security.

C. Challenge: Risk reduction

1) Recommendation

Financial institutions and e-Commerce merchants must per- form specific risk analysis for their environments; taking in- to consideration actual loss, number of incidents, number of customers involved and vulnerabilities of the available authentication methods. Based on such analysis, choose the eIDAMs that effectively reduce the number of incidents and loss.

2) Recommendation

implementing a context-based authentication strategy can highly increase the confidence level and the overall security regarding user authentication and consequently reduce the number of incidents and loss.

3) Recommendation

the concept of “something the user has” can be extended to the IP address, device and platform used to access e- Banking services; therefore, it is recommended to register users’ device, platform and even mobile applications used by customers and verify them in continuous authentication eIDAMs. Real time validations of the authenticity of such parameters should be required in order to reduce risks.

4) Recommendation

Static analysis of the security of users’ devices used to access e-Banking services should be performed and evaluated directly or indirectly. Result of this analysis could be used to trigger the requirement of additional eIDAMs.

5) Recommendation

eIDAMs to be implemented may include elements linking authentication to the specific amount and destination account by providing non-repudiation, end-users may feel more confident with the transactions they authorize.

CHAPTER 7: RESULTS & CONCLUSIONS

In the Malware research in chapter 4th we have showed a complete solution to detect certain families of Zeus, one of the most dreadful financial mal-wares. In particular, we have exposed a technique that allows to extract the keystream used by Zeus to encrypt its payload. Based on these results, an IDS to detect Zeus (*Cronus*) has been detailed, and it has been experimentally tested on a production network. Excellent performance and effectiveness results achieved by *Cronus* support our findings. Finally, we have reported on lesson learning and highlighted future work.

As shown in chapter 5th he mobile App-Stores should ensure they take into consideration the current security issues, including the known threats and vulnerabilities when permitting an App to be published on their stores, in order to enhance the overall security of the App delivery management mechanism.

Similarly, the Apps developers, while developing an App, should not only consider an App's functionality and usability but must also take into account the security aspects of that specific App. This could include App isolation, ensuring that an App must not attempt to access unnecessary resources, etc.

Research shown in chapter 6th presented a thorough study of current practices and security challenges associated to the implementation of eIDAMs in electronic banking. Based on our analysis, it can be concluded, that there is no one-fits-all approach when implementing secure identification and authentication. This is a result of the fact that not all financial institutions may cover one or more user segments and/or operation types; therefore, a different set of requirements is needed by each of them, affecting also the selection criteria that must be used. For that reason, it is strongly recommended that security professionals perform risk and cost-benefit analyses of their e- Banking ecosystem, in order to able to better decide on the most suitable eIDAM(s) to be implemented.

Chapter 6th has also identified that due to the rapid evolvement of technologies, the emerging trends in the e-Banking security area are now focusing on mobile and biometrics based authentication. The combination

of both, mobile and biometrics are promising state-of-the-art solutions, providing high level of security allowing the evaluation of context information by analyzing the users' behavior and identifying user devices and platforms used to access the service.

Future research directions will provide the design principles and experimentation of an authentication system based on biometrics deployed in real production environment, which, integrates continuous and transaction authentication. The adoption of cloud-based authentication has not been discussed in this research, mainly because it is not really considered an eIDAM, but a form of implementing it. Nevertheless, cloud-based authentication promises reduction of complexity and ease of scalability, for this reason it is worth to be explored in future research.

It has been shown that by analyzing Malware samples we can characterize their behavior in control devices for accessing online banking services to identify customers when they supplant the legitimate user.

This method can detect not only Family ZEUS malware Trojans, but also those families that use a similar system of encryption. This technology has been used by different families of malware during the period in which the scientific study is done.

Detection of illegal access by this method has been shown to be efficient and has detected several ZEUS malware families, and several waves of attacks against the online banking in Caixabank.

The proposed recommendations for mobile Apps have been effective to prevent inappropriate use of unsafe stores like focus of malware infection.

The methods of two-factor authentication used by European banks as a general concept are currently effective in controlling the Zeus malware families. Despite the fraud supported by the banking sector in this period attributable to this malware without strong authentication methods, using dual factor only (pin 1 and pin2, etc.) amounts would certainly be much higher if they have not turned the authentication to more strong methods. The results obtained through these work helped entities to take decisions on what strong methods implement in e-banking to avoid large fraud amounts.

Since the greatest amount of fraud have supported those entities that did not use double factor the banking industry turned to change and implemented authentication with robust methods; while those entities using these robust, strong methods (OTP, OTP/SmS, etc.) have been markedly less economically disappointed.

Biometrics techniques used in the scientific study are varied and their application as double factor is still controversial. The POC conducted show that the technologies used as identification by biometric techniques to implement the two-factor required by the ECB still needs to improve the algorithms to implement in terms of usability that are better accepted by the users (customers). This is particularly suitable in facial identification technology.

ANNEX 1 SCIENTIFIC PUBLICATIONS

In the following table are the publications, presentations and proceedings issued from the research works related to the thesis:

2011

- eCRS IEEE 11 - Riccardi, M., Pietro, RD, & Vila, JA (2011, November).Taming Zeus by leveraging its own crypto internals. In/eCrime Researchers Summit (eCrime), 2011/(pp. 1-9).IEEE

2012

- IEEE12M Z. Kazmi, T. Felguera, J. Aguila, M. Maawad, "TASAM – Towards the smart devices App-Stores Applications security Management related best practices," at the [NTMS 2012] IEEE's 5th International Conference on New Technologies, Mobility and Security, Istanbul, Turkey, May 2012 - E-ISBN: 978-1-4673-0227-2

2013

- CN 13 Riccardi, M., Di Pietro, R., Palanques, M., & Vila, J. A. (2013).Titans' revenge: Detecting Zeus via its own flaws./Computer Networks/,/57/(2), 422-435
- WID 13 -J. Aguila, J. Serna-Olvera, M. Medina, A. Sfakianakis and L.A. Fernández, "Mobile Banking Authentication: Assessing the Robustness of Authentication Mechanisms", in Proceedings of the World eID Congress, September 2013, Nice, France. Online available: http://www.worlde-idcongress.com/proceedings
- ISSE 13 – J. Aguila, M. Medina, "e IDAS Landscape for e Banking Services", Proceedings of ISSE, 22-23 Oct. 2013, MCE Brussels, Belgium.
- ENISA 13 -M. Medina, J. Aguila, J. Serna-Olvera, A. Sfakianakis and L.A. Fernández, "eIDAS in e-Finance and e-Payment services - Current practices and Recommendations", ENISA Report, December 20 2013, ISBN: 978-92-9204-077-2 doi: 10.2824/27272. Disponible Online : https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/eIDAS-in-e-finance-and-e-payment-services
- IEEE 13 -Aguilà VILA, J.; Serna J.; Medina M., Stefaniakis, A., Fernández, L.A. " A professional view on eBanking Authentication Challenges &Recommendations" 9th International Conference on Information Assurance &Security. Yassmine Hammamet, Tunisia. 4-6 Ec. 2013

2014

- JIAS 14 – J.Aguilà Vilà, J.Serna, M. Medina and A. Sfakianakis, " An Analysis of n factor Authentication in e-Banking environments", Journal of Information Assurance & Security , ISSN 1554-1010 Volume 9 (2014) pp 104, 117. @MIR labs, www.mirlabs.net/JIAS/index.html

Research works are published in articles that focus Malware and cases of use of the Zeus Trojan family, the articles published on authentication methods, Malware barriers that must be overcomed to reach the money and the work carried out as a review of security for mobile platforms (Table 1 below)

CODIGO	TÍTULO PUBLICACIÓN	PUBLICADO
eCRS-11/IEEE11	(1) "Taming Zeus by leveraging its own crypto internals"	IEEE: 2011 ISSN 2159-1237 E-ISBN: 978-1-4673-0227-2 PRINT ISBN 978-1-4577-1338-5 DOI:10.1109/eCrime.2011.6151981
CN-13	(2) "Titan's revenge: detecting Zeus via its own flaws"	CN: 2013 Computer Networks [online] ISSN 1389-1286
IEEE12M	(3)"TASAM – Towards the smart devices App-Stores Applications security Management related best practices"	IEEE: 2012 ISSN 2157-4952 E-ISBN: 978-1-4673-0227-2 PRINT-ISBN 978-1-4673-0228-9 DOI 10.1109/NTMS. 2012.6208729
ENISA 13	(4) "eIDAS in e-finance and e-payment services-Current practices and Recommendations"	ENISA: 2013 Print ISBN: 978-1-4799-2989-4 DOI: 10.1109/ISIAS.2013.6947731
IEEE-13	(5) "A professional view on eBanking Authentication Challenges & Recommendations"	IEEE/IAS: 2013 Print ISBN: 978-1-4799-2989-4 DOI: 10.1109/ISIAS.2013.6947731 **Premio a la mejor publicación del congreso**
JIAS-14	(6) "An Analysis of n-factor Authentication in e-Banking Environments"	JIAS: 2014 Journal of Information Assurance and Security. ISSN 1554-1010

Table 1: Summary of publications

the proceedings and presentations from Table 2 are directly related and part of the body of research. Although they have not been published in magazine or publications are important contributions to the research

CODIGO	TÍTULO	CONGRESO
WID-13	(7) "Mobile Banking Authentication: Assesing the robustness of authentication mechanisms"	PRESENTACIÓN WORLD e ID
ISSE-13	(8) "eIDAS Landscape for e-banking Services"	PROCEEDINGS ISSE 2013

Table 2: "Proceedings" and conference presentations:

Nomination and statement of previous works and publications related to the research:

At the beginning of the research in e-crime projects run by me, other works were published "benchmarking ip blacklist for financial botnet detection" presented at the "Sixth International Conference on Information Assurance and Security IAS2010 Conference, Atlanta, USA 2010 Authors D. Gold., J. Moon., A. Felguera., M. Vilanova., and J. Serna. The paper "A Framework For Financial Botnet Analysis" presented in "The eCrime Researchers Summit 2010 (ECRS 2010) Dallas, Texas, authors M. Riccardi., D. Gold., J. Moon and M. Cremonini.

These previous studies were performed with mixed teams of Barcelona Digital Foundation and my team at Caixabank, with the aim to better understand the scientific environment of the called "botnets" systems, before entering the research on authentication systems and Malware (Trojans) that rely on this type of infrastructure.

The work presented in Texas won the IEEE-SA APWG eCrime fighters Scholarship Award.

These works are not included in the thesis publications list, since I did not participate in them as an author but as director of the research project, but I nominate its existence because they quote and are important for new researchers because are directly related to the research of authentication and Malware and because they were developed in the environment research projects that I lead and coordinate since 2010.

The research areas field related to e-crime is very broad and there are different varied lines of research. My research included financial institutions online and major threats, response or defenses that are at the cutting edge, whether real and presenting threats to industry bank and its customers, (eg The latest financial malware family, the latest version of the family of a very active Trojan Zeus against the bank, types of attacks against mobile client systems connecting with the bank, etc. Authentication systems, use of biometrics authentication systems, state of the art in mobile

applications, research on the safety of the platforms that connect to online banking (mobile phones, ipads, tablets, etc.)

Focus in the works and their context, geographically speaking, if anything this distinction is possible on the Internet, we have chosen threats affecting the environment of banks in Europe, in a European customer, which does not mean at all that, could not affect other geographical regions of the world to being global threats that do not distinguish borders. That is why we have developed several studies on families financially Malware and Trojan called Zeus (such as use case Malware), about the safety of mobile applications on the systems authentication and security level banks, on use of various technologies in the authentication methods in the mobile security. APP's and their developments made to improve the response of the security groups (CSIRTS, CERTs), etc.

I have also made a number of works both forensics and electronic research in general; the results have not been published due to confidentiality of results or the environment in which they are incurred.

Public research carried out in the environment of e-crime begins in 2010. My role as director of the research projects in "La Caixa" has always been intervening directly deciding the topic, focus and objectives of the research and directing research and research funding in all cases.

In the development of the research I have been directly involved either in person and working with my team members as "senior" or in decisions of research projects or in the development of the projects themselves.

Due to interest to us, the financial sector, authentication systems and the threat to these systems embodied in malware and field research Trojans, we have published works that I have done research tasks directly, I have also directed directly as first author all work the research.

The research results of the thesis show a progression that includes the work of the line intrinsically related to authentication methods and Malware research. Each of the research topic and the correspondent publications represents a progression or a study of a pending important aspect in the earlier work of the follow consistent and research publications.

Publication: ECRS-11 Code / IEEE 11

(1) Taming Zeus by leveraging ITS own crypto internals.

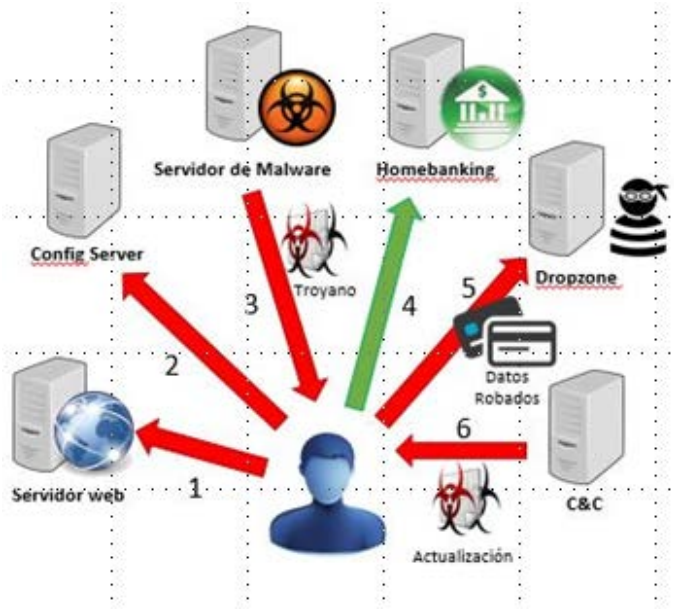
The contribution of the first publication of the Zeus malware "Taming Zeus by leveraging internals crypto Its Own" can be summarized in the following points:

a) First, the scheme and operation of malware infection is described: The following diagram shows the steps from a user is infected until their data is captured and sent to a server (C & C) controlled by the owners of the botnet Zeus Trojan.

Explanation of the steps.

1. The user, or surfing the net or clicking on a link in an email, access to a website where it is infected by an "exploit kit" that takes advantage of the navigation software vulnerability of the victim to inject malicious code , usually a "dropper" (Downloader malware)

2. The "dropper" is running on the victim machine, connect to a second server, and download the banking Trojan, that is prepared for stealing user data.



3. Once the Trojan has been downloaded and installed on the victim machine, this request to download the configuration file. The file can be found in different locations (in the public forums, in a compromised server, etc.) and is encrypted in this case of use, using an encryption system based on RC4 plus XOR operations. This configuration file contains instructions and details of the entities affected by this malware.

4. The user connects to the website financial institution, the malware that is monitoring the user activities is activated when the user navigates to the website of the bank and has specific instructions to capture data access and authorization operations.

5. The user sends the data, believing that is only sent to the bank, but actually is also sent to a server controlled by the criminal infrastructure. If the malware needs some updating, its mode of operation will gradually time to time communicate with Command & Center of the botnet (C & C) to receive updates.

b) Second, the authors propose a general methodology to break the encryption communications malware.

c) Third, we provide a proof of concept of this methodology applied in the production environment.

d) Fourth, we showed that this methodology can be used to respond to the threat of Zeus and finally highlighting lessons learned we provided some general principles of Malware (in general) and Zeus in particular.

As we perform future works the we will propose a generalization of the keystream technique and propose some IDS defense system

Publication: Code CN-13

(2) "revenge Titans: Zeus detecting flaws via Its Own"

This broad line of research work started with the previous work and complete the remaining tasks in the research, introducing recommendations to counter attack the malware analyzed in the above publication.

The contributions of this publication are:

a) First, the authors propose a technique to break communications malware (keystream) extracting the system used to encrypt such communications.

b) Second, we propose a generalization of the keystream technique

c) Third, we propose Cronus, an IDS that specifically seeks the Zeus malware, testing it experimentally in a production network and their skills and effectiveness are discussed.

d) Finally we highlight some principles that reside in the malware- and Zeus in particular which focus future research in this field.

In both works (1) and (2) it should be noted that the version of the Trojan is the latest at the time of the malware family with what was the most recent of all the papers published so far, so this research It is a novelty contribution to the scientific world.

To Perform outstanding work we will make a review of authentication systems to prevent the spread and infection by malware and as a means of prevention. Make a State of the art of the authentication methods in online banking. And a research study of the art mobile apps as a way of malware infection.

Publication: Code IEEE12M

(3) "Tasam - Towards the App-Stores smart devices Security Management Applications related best practices".

In this publication a review of the safety of mobile applications is performed as downloads Apps are used for financial services such as discharge Malware point and hide trojans deceiving customers with social engineering techniques, leading to a false application (the Trojan) that allows the attacker to obtain credentials and perform fraud to ebanking customers.

The publication introduces the following scientific contributions:

a) First is a review of the main app stores for mobile financial services.

b) Second, it offers a proposal for these stores and provides a set of best practices and recommendations for improving safety management based on the review of outstanding work performed: Concept tests on mobile platforms, POC authentication methods and authentication methods. Open mobile banking. Safety tests on mobile devices. Reinforce the management of security and methods on mobile devices.

Publication ENISA Code 13

(4) "eIDAS in e-Finance and e-Payment services - Current Practices and Recommendations"

"EIDAS in e-Finance and e-Payment services - Current Practices and Recommendations" is a report that summarizes the results of the survey elaborated by me and released by ENISA (European Network and Information Security Agency) and APWG.eu. The main objective of the survey was to gather information about systems eIDAS systems used in e-finance and e-payment. Analyzes the risk associated with each authentication mechanism implemented in the financial sector and recommends lines of action and best practices for the major players in this sector and gathering the results.

In this research the tasks performed are described for consistent research work in developing the wide survey that touches all the various aspects related to the use of eIDAS systems; Segments, channels, systems, mode of use, place of use, time of use, appliances, software, reliability, usability, resulting from their use, economic consequences of its use, ect. We perform tasks to assemble a sufficient database of contacts that conduct the survey, as well as for publication on the website of ENISA and about 120 respondents professionals CISO's, CIO's COO's banking sector.

The contributions of "eIDAS in e-Finance and e-Payment services - Current Practices and Recommendations" are:

a) First: Collect survey results released by ENISA (European Network and Information Security Agency) and APWG.eu.

b) Second: The main objective of the survey was to gather information about systems eIDAS systems used in e-finance and e-payment.

c) Third: Analyze the risk associated with each authentication mechanism implemented in the financial sector and recommends lines of action and best practices for the major players in this sector.

Pending jobs to do: identify and plan proof of concept of the recommendations made.

Publication IEEE-13 Code

(5) "A professional view on eBanking Authentication: Challenges and Recommendations"

This work is where first introduced the issue of biometrics analyzing the reasons why it has not been implemented as a factor of authentication in most of the surveyed entities. In "A professional view on eBanking Authentication: Challenges and Recommendations" common threats in e-banking are analyzed as well as the current state of art, analyzing the most popular eIDAS systems implemented in Europe.

Identifies threats and analyzes the latest attack scenarios in the environment of e-banking authentication. Analyzes the most common aspects of eIDAS recommended for electronic banking and its usefulness to most popular threats in terms of related incidents financial losses. Also recommend on solutions to consider to choose and implement eIDAS systems against threats (phishing, Trojans, etc.)

Since it is based on responses from professionals taking decisions on their systems used by banks, and its customers, the authors, j eagle. J. Serna, L.A. Fernandez, M. Medina, A. Sfakianakis present a updated vision and a very real recommendations on various aspects eIDAS. Regarding the previously published work novelty is that this work is based on eIDAS systems through responses from professional and not based on mere observation of their use on the websites of the companies analyzed.

This work received the award for the best publication of congress 9th International Conference on Information Assurance and Security (IEEE) in Tunisia.

The following scientific contributions are made:

a) First, common threats in e-banking are analyzed as well as the current state of art, analyzing the most popular eIDAS, systems implemented in Europe, through a survey elaborated by us, launched by ENISA, APWG.eu, and FS-ISAC, and answered for security professionals of the financial sector in Europe.

b) Second: It identifies threats and analyzes the latest attack scenarios in the environment of e-banking authentication.

c) Third: Analyze the most common aspects of eIDAS recommended for electronic banking and its useful/usefulness to the most popular threats in terms of financial losses. Also it recommends on solutions to consider at the moment to choose and implement eIDAS systems.

d) Fourth: Since it is based on responses from the professionals who make decisions about the systems used by its banks, (Aguilà J. et al) present a very updated vision and a very real recommendations on various aspects eIDAS.

e) Fifth: Regarding previously published work to this line of research novelty is that this work is based on eIDAS systems through responses from professional and not based on mere observation of use in pages web of institutions analyzed.

Pending jobs to do: identify and plan proof of concept of the recommendations made. Propose new models of multimodal identification.

Publication: JIAS-14 Code

(6) "An analysis of n-factor Authentication in e-banking Environments

This paper provides an overview of potential threats in the use of electronic banking. Analyzes the most common authentication (eIDAMs) methods in the European financial sector, giving an overview of its robustness against electronic attacks.

It’s also conducting an analysis of adequacy as operating systems and factors like usability and cost efficiency It also introduces the possibility of using biometric authentication methods.

This paper presents the first results of proof of concept identification using unconventional methods such as biometrics that are introduced.

This publication provides the following contributions:

a) First provides a general situation and the impact of current major threats to online banking.

b) Second analyzes the most common methods eIDAMS ; authentication methods and electronic identification, implemented in Europe and robustness.

c) Third presents an analysis of adequacy in terms of efficiency, usability, costs, types and operating segments.

d) Includes an analysis of possibilities with biometrics as innovation. Propose new models of multimodal identification. POC perform with biometrics on mobile.

Pending jobs to do: identify and plan more wide proof of concept of the recommendations made.

We realized other conference proceedings and presentations:

Presentation World and ID:

(7) "Mobile Banking Authentication: assessing the robustness of authentication mechanisms"

The contributions of "Mobile Banking Authentication: assessing the robustness of authentication mechanisms" are:

a) First, analyzes the current threats in mobile banking applications.

b) Second, identifies the most popular authentication mechanisms on mobile devices.

c) Third, advises on the robustness of these mechanisms against known threats by providing various metrics in terms of occurred incidents

d) Fourth, finally, introduces a set of recommendations concerning countermeasures that should be considered in any authentication system

.

Perform outstanding work: Review of eIDAS on various platforms and concept testing.

Proceedings ISSE 2013

(8) "eIDAS landscape for ebanking services"

The contribution of the paper presented at ISSE in Brussels "eIDAS landscape for ebanking services" is:

a) First, summarizes the results of the survey conducted in collaboration with ENISA, and

b) Second: it presents the first recommendations for improving the selection methodology of eidas methods.

Pending jobs to do: identify and plan proof of concept of the recommendations made.

ANNEX 2. Bibliography

[1] M. Chandrasekaran, Chinchani R., and S. Upadhyaya, "Phoney: Mimicking user response to detect phishing attacks," in Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks. IEEE Computer Society, 2006, pp. 668-672.

[2] D. Birk, S. Gajek, F. Grobert, and A. Sadeghi, "A forensic tracing framework for phishers," IFIP Summer School on The Future of Identity in the Information Society, Karlstad, Sweden, 2007.

[3] L. Spitzner, "Honeytokens: The other honeypot," Security Focus, vol. 21, 2003.

[4] P. Bacher, T. Holz, M. Kotter, and G. Wicherski, "Know Your Enemy: Tracking Botnets" The Honeynet Project, 2005.

[5] M. Cremonini and M. Riccardi, "The Dorothy Project: An Open Botnet Analysis Framework for Automatic Activity Tracking and Visualization," in Proceedings of the 5th European Conference on Computer Network Defense (EC2ND). IEEE, 2010, pp. 52-54.

[6] M. Riccardi, D. Gold, J. Moon, M. Cremonini, and M. Vilanova, "A framework for financial botnet analysis," in eCrime Researchers Summit (eCrime), 2010. IEEE, 2011, pp. 1-7.

[7] J. Caballero, NM Johnson, S. McCamant, and D. Song, "Binary code extraction and interface identification for security applications," in In ISOC NDSS'10, 2010.

[8] F. Leder, P. Martini, and A. Wichmann, "Finding and extracting crypto routines from malware," in Performance Computing and Communications Conference (IPCCC), 2009 28th IEEE International. IEEE, 2009, pp. 394-401.

[9] F. Leder and P. Martini, "Ngbpa next generation protocol analysis botnet," Emerging Challenges for Security, Privacy and Trust, pp. 307-317, 2009.

[10] Y. Kim and H. Youm, "A new bot disinfection method based on DNS sinkhole," Journal of KIISC, vol. 18, pp. 107-114, 2008.

[11] S. Ji, C. Im, M. Kim, and H. Jeong, "Botnet detection and response architecture for Offering secure internet services," in Security Technology, 2008. SECTECH'08. International Conference on. IEEE, 2008, pp. 101-104.

[12] Y. Kim, D. Lee, J. Choi, and H. Youm, "Preventing damage botnet technique and Its bot using dns sinkhole effect," Journal of KISS (C): Computing Practices, vol. 15, no. 1 pp. 47-55, 2009.

[13] G. Gu, P. Porras, Yegneswaran V., M. Fong, and W. Lee, "BotHunter: Detecting malware infection driven dialog ids through correlation," in Proceedings of 16th USENIX Security Symposium USENIX Security Symposium on. USENIX Association, 2007, p. 12.

[14] X. Jiang, X. Wang, and D. Xu, "Stealthy malware detection Through VMM-based out-of-the-box view semantic reconstruction," in Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, 2007, pp. 128-138.

[15] T. Ormerod, L. Wang, M. Debbabi, A. Youssef H. Binsalleeh, Boukhtouta A., and P. Sinha, "Defaming botnet toolkits: A bottom-up approach to mitigating the threat," in 2010 Fourth International Conference on Emerging Security Information, Systems and Technologies.IEEE, 2010, pp. 195-200.

[16] R. Ford and S. Gordon, "Cent, five cent, ten cent, dollar: hitting botnets Where it really hurts," in Proceedings of the 2006 workshop on New security paradigms. ACM, 2006, pp. 3-10.

[17] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang, "On the analysis of the botnet Zeus crimeware toolkit," Trust in Security and Privacy (PST), 2010 Eighth Annual International Conference on. IEEE, 2010, pp. 31-38.

[18] D. Marcus and R. Sherstobitoff, "Dissecting Operation High Roller," McAfee, 2012

[19] D. B. Were Kalige, "A Case Study of Eurograbber: How 36 Million Euros via Malware was Stolen," Versafe (White paper), 2012.

[20] M. R. Nami, "E-Banking: Issues and Challenges," in 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel / Distributed Computing, 2009.

[21] C. K. Dimitriadis, "Analyzing the Security of Internet Banking Authentication Mechanisms," Information Systems Control Journal, 2007.

[22] L. Peotta, MD Holtz, BM David, FG Deus and RT d. S. Jr., "A Formal Classification of Internet Banking Attacks and Vulnerabilities," International Journal of Computer Science & Information Technology, pp. 186-196, 2011.

[23]. Fatima, "E-Banking Security Issues - Is There A Solution in Biometrics ?," Journal of Internet Banking and Commerce, August 2011, Vol. 16, no.2, vol. 16, no. 2, 2011.

[24]. Hiltgen, T. and T. Krampand Weigold, "Secure Internet Banking Authentication," IEEE Security & Privacy, 2006.

[25] R. Chouhan and VS Rathore, "e-Banking Security and Authentication Issues," Referred International Research Journal, 2011.

[26] J. Choubey and B. Choubey, "Secure User Authentication in Internet Banking: A Qualitative Survey," International Journal of Innovation, Management and Technology, vol. 4, no. 2 2013

[27] SS Corporation and M. Ligh, "[PRG] malware case study," Secure Science Corporation, Tech. Rep., 2006.

[28] R. Howard, Cyber Fraud: Tactics, Techniques and Procedures, 1st ed. Boston, MA, USA: Auerbach Publications, 2009.

[29] (2009) Top-10 botnet Outbreaks in 2009. [Online]. Available: http: // blog. damballa.com/?p=569

[30] (2010) Banking Zeus malware successfully bypasses anti-virus detection. [Online]. Available: http://www.trusteer.com/company/press/trusteerwarns- Zeus Trojan-bypasses-date-anti-virus-systems-77-percent-time

[31] (2011) Zeus source code for sale. got 100,000 dollars? [Online]. Available: http://krebsonsecurity.com/2011/02/zeus-sourcecode-for-sale-got-100000/

[32] (2011) Zeus tracker. [Online]. Available: https://zeustracker.abuse.ch/statistic.php

[33] (2011) Google translate service. [Online]. Available: http://translate.google.com/

[34] J. Manuel, Another modified zeus variant seen in the wild [cited 19/12/2011].
URL http://blog.trendmicro.com/another-modified-zeus-variant-seen-in-the-wild/?awid=7917255160271489866-1985

[35] D. Kahn, The Codebreakers: the story of secret writing, Scribner Book Company, 1996.

[36] Goodin, D. (2011), Malicious apps infiltrate Google's Android Market. Available at: http://www.theregister.co.uk/2011/12/12/android_market_malware/ [Accessed, 12th December 2011]

[37] Zetlin, M. (2011), 8 Tips to Stop Banking App Fraud. Available at: http://www.creditcards.com/credit-card-news/eight-tips-stop-banking-app-fraud-1282.php [Accessed, 24th August 2011]

[38] comScore Data Gem, "1 in 4 Internet Users Access Banking Sites Globally" 2012. [Online] Available: https://www.comscore.com/Insights/Data-Mine/1-in-4-Internet-Users-Access-Banking-Sites-Globally. [Last accessed Sept 2014].

[39] N. Bhas, "Juniper Research", 2013. [Online] Available: http://www.juniperresearch.com/viewpressrelease.php?pr=356. [Last accessed Sept 2014].

[40] Ernst & Young, "EY Global Consumer Bank- ing Survey 2014" 2014. [Online]. Available: http://www.ey.com/Publication/vwLUAssets/EY-- Global Consumer- Banking Survey 2014/$FILE/EY- Global-Consumer-Banking-Survey-2014.pdf.[Last accessed Sept 2014].

[41] European Commission, "Special Eurobarometer 404: Cyber Security" November 2013. [Online]. Available: http://ec.europa.eu/public opinion/archives/ebs/ebs 404 en.pdf. [Last accessed Sept 2014].

[42] S. Kiljana, K. Simoens, D. De Cock, M. van Eekelen and H. Vranken, "Technical report: Security of Online Banking Systems" 2014.

[43] D. Marcus and R. Sherstobitoff, "Dissecting Operation High Roller", McAfee, 2012.

[44] D. B. Eran Kalige, "A Case Study of Eurograbber: How 36 Million Euros was Stolen via Malware", Versafe (White paper), 2012.

[45] M. R. Nami, "E-Banking: Issues and Challenges", in *10th ACIS International Conference on Software Engi- neering, Artificial Intelligences, Networking and Paral- lel/Distributed Computing*, 2009.

[46] C. K. Dimitriadis, "Analyzing the Security of Inter- net Banking Authentication Mechanisms", *Information Systems Control Journal*, 2007.

[47] L. Peotta, M. D. Holtz, B. M. David, F. G. Deus and R. T. d. S. Jr, "A Formal Classification of Internet Banking Attacks and Vulnerabilities", *International Journal of Computer Science & Information Technology*, pp. 186- 196, 2011.

[48] A. Fatima, "E-Banking Security Issues C Is There A Solution in Biometrics?", *Journal of Internet Banking and Commerce*, August 2011, vol. 16, no.2, vol. 16, no. 2, 2011.

[49] S. Samine Hosseini and S. Mohammadi, "Review Banking on Biometric in the World's Banks and Intro- ducing a Biometric Model for Iran's Banking System", Journal of Basic and Applied Scientific Research, vol. 2, no. 9, 2012

[50] A. Hiltgen, T. Krampand and T. Weigold, "Secure Inter- net Banking Authentication", *IEEE Security & Privacy*, 2006.

[51] R. Chouhan and V. S. Rathore, "e-Banking Security and Authentication Issues", *International Referred Re- search Journal*, 2011.

[52] J. Choubey and B. Choubey, "Secure User Authentica- tion in Internet Banking: A Qualitative Survey", *Inter- national Journal of Innovation, Management and Tech- nology*, vol. 4, no. 2, 2013.

[53] R. Tassabehjia and M. A. Kamalab, "Evaluating bio- metrics for

online banking: The case for usability," In- ternational Journal of Information Management, vol. 32, p. 489C494, 2012.

[54] A. Eng and L. A. Wahsheh, "Look Into My Eyes: A Survey of Biometric Security," in IEEE Conference International Conference on Information Technology: New Generations, 2013.

[55] Petrovska-Delacrtaz, Dijana, Chollet, Grard, Dorizzi and Bernadette, "Guide to Biometric Reference Sys- tems and Performance Evaluation," 1 ed., Springer Pub- lishing Company, Incorporated, 2009.

[56] P. Shen Teh, A. Beng Jin Teoh and S. Yue, "A Survey of Keystroke Dynamics Biometrics," The Scientific World Journal, vol. 2013, 2013.

[57] Division of Consumer and Community, "Consumers and Mobile Financial", Board of Governors of the Fed- eral Reserve System, 2014.

[58] A. Y. Lindell. "Time versus Event Based One- Time Passwords", Aladdin Knowledge Systems Ltd. [Online] Available: http://www3.safenet- inc.com/blog/pdf/Time vs Event Based - OTP.pdf. [Last accessed Sept 2014].

[59] imacros plugin. [Online]. Available: http://www.iopus.com/imacros

[60] comScore Data Gem, "http://ww.comscoredatamine.com/," 2012. [Online]. Available: 1 in 4 Internet Banking Users Access Sites Globally. [Accessed July 2013].

[61] N. Bhas, "Juniper Research," 2013. [Online]. Available: http://www.juniperresearch.com/viewpressrelease.php?pr=356. [Accessed July 2013].

[62] Division of Consumer and Community, "Consumers and Mobile Financial," Board of Governors of the Federal Reserve System, 2013

[63] A. Y. Lindell. [Online]. Available: http://www3.safenet-inc.com/blog/pdf/Time_vs_Event_Based_OTP.pdf. [Accessed 21 July 2013]

[64](2011)Annuncing pcapr-premises.[Online].Available: http://blog.mudynamics.com/2011/04/18/announcing-pcaprlocal/