



TRABALHO DE GRADUAÇÃO

Análise de Artefatos Maliciosos em Ambiente Acadêmico

Alysson de Sousa Ribeiro

Walisson Francisco de Albuquerque

Brasília, Dezembro de 2014

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

Análise de Artefatos Maliciosos em Ambiente Acadêmico

Alysson de Sousa Ribeiro

Walisson Francisco de Albuquerque

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação*

Banca Examinadora

Prof. Laerte Peotta Melo, ENE/UnB
Orientador

Prof. João Gondim, CIC/UnB
Examinador interno

Dr. Dino Macedo Amaral, Banco do Brasil
Examinador externo

Dedicatórias

A todos que acreditaram nesse projeto e apoiaram de alguma forma. Obrigado.

Walisson Francisco de Albuquerque

A todos que contribuíram para a realização deste trabalho. Muito obrigado.

Alysson de Sousa Ribeiro

Agradecimentos

Gostaria de agradecer à minha mãe Fátima, ao meu pai Adonato e ao meu irmão Andersson pela minha formação como pessoa, sem vocês este trabalho não seria possível. À minha namorada Aline por todo companheirismo desde que nos conhecemos e pela sua ajuda e paciência nos momentos importantes desse projeto. Aos colegas Tito e Renato por iniciarem esse trabalho e por todo auxílio na implementação do projeto. Aos amigos e parentes que estiveram comigo nesse período, em especial ao Leonardo por todo auxílio profissional que me passou. Aos colegas do CPD/UnB, em especial ao Karam e ao Domingos, pela estrutura cedida para hospedagem desse trabalho. Ao professor Peotta e ao meu colega Walisson por estarem juntos comigo em toda a trajetória desse projeto.

Alysson de Sousa Ribeiro

Agradeço primeiramente a Deus, pois sem Ele eu não teria condições de chegar até aqui. Agradeço à minha mãe Edna, meu pai José e minha irmã Flávia, pois sempre foram a base para a realização de todos os meus sonhos e planos, me apoiando sempre. Agradeço à minha namorada Greicy, que me apoiou incondicionalmente, auxiliou-me e motivou-me a continuar, mesmo em meio a adversidades. Agradeço aos professores e funcionários do Departamento de Engenharia Elétrica, que contribuíram muito ao longo de minha formação. Agradeço aos amigos do CPD/UnB pelo apoio e ensino, especialmente Karam e Domingos, pelas diversas oportunidades que me proporcionaram e pelo conhecimento que me passaram. Agradeço ao professor Laerte Peotta, que aceitou o desafio de me orientar e auxiliou bastante em todas as etapas do trabalho. Agradeço ao meu colega e amigo Alysson, pois esteve comigo desde o início do trabalho e me auxiliou e ensinou ao longo do projeto. Muito obrigado!

Walisson Francisco de Albuquerque

Resumo

Análise de Artefatos Maliciosos em Ambiente Acadêmico

Autor: Alysson de Sousa Ribeiro e Walisson de Albuquerque

Orientador: Laerte Peotta

Projeto Final de Graduação em Engenharia de Redes de Comunicação

Brasília, Dezembro de 2014

O número e a variedade de códigos maliciosos vêm crescendo de forma rápida. Esses códigos estão cada vez mais sofisticados, incorporando diversas técnicas para dificultar a identificação de suas ações, tornando a análise de malwares uma ferramenta importante no combate aos crimes virtuais.

A análise de um artefato visa compreender suas características, quais ações serão realizadas e como será executado no sistema operacional. Esse projeto apresenta a implementação do UnBox, uma ferramenta que utiliza diversas técnicas de análise para facilitar a identificação de um malware.

O UnBox está hospedado no Centro de Informática da Universidade de Brasília (CPD/UnB), conta com uma interface web em português para o envio de artefatos e URLs e para a apresentação dos relatórios das análises.

Palavras-chave: Análise Dinâmica, Análise Estática, Malware, UnBox, Códigos Maliciosos, Cuckoo Sandbox.

Abstract

The number and variety of malicious code have been growing quickly. These codes are increasingly sophisticated, incorporating various techniques to hinder the identification of their actions, making malware analysis an important tool in the fight against cybercrime.

The analysis of an artifact aims to understand its characteristics, what actions will be taken and how it will be run on the operating system. This design presents the implementation of UnBox, an analysis tool that uses various techniques to facilitate identification of malware.

The UnBox is hosted in Informatics at the University of Brasilia Center (CPD / UNB), has a web interface in Portuguese for sending articles and URLs and for the presentation of the analysis reports.

Keywords: Malware, analisys, UnBox, malicious code, Cuckoo sandbox.

SUMÁRIO

RESUMO	I
ABSTRACT	II
LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES	VIII
1 INTRODUÇÃO	1
1.1 OBJETIVOS	1
1.2 OBJETIVOS ESPECÍFICOS	1
1.3 JUSTIFICATIVA	2
1.4 ORGANIZAÇÃO DO TRABALHO	2
2 FUNDAMENTAÇÃO TEÓRICA	4
2.1 SPAM E <i>Phishing</i>	4
2.2 CÓDIGOS MALICIOSOS	5
2.2.1 VÍRUS	6
2.2.2 WORMS	6
2.2.3 <i>Bot</i> e <i>botnet</i>	7
2.2.4 CAVALO DE TRÓIA (<i>trojan</i>)	8
2.2.5 <i>Spyware</i>	8
2.2.6 <i>Rootkit</i>	9
2.2.7 <i>Backdoor</i>	9
2.3 ANÁLISE DE MALWARE	9
2.3.1 ANÁLISE ESTÁTICA	10
2.3.1.1 ANÁLISE DE CÓDIGO	10
2.3.1.2 FUNÇÃO DE <i>Hash</i>	10
2.3.1.3 ANÁLISE DE <i>Strings</i>	11
2.3.1.4 BIBLIOTECAS E FUNÇÕES UTILIZADAS	11
2.3.1.5 OFUSCAMENTO DO CÓDIGO	12
2.3.1.6 ANTIVÍRUS	13
2.3.2 ANÁLISE DINÂMICA	13
2.3.2.1 <i>Cuckoo Sandbox</i>	14
3 DESENVOLVIMENTO	15

3.1	UNBOX: FRAMEWORK DE ANÁLISE	15
3.1.1	FUNCIONAMENTO DO UNBOX	16
3.1.2	ETAPAS DE ANÁLISE DO UNBOX.....	17
3.1.3	MÓDULOS DO UNBOX.....	18
3.1.4	YARA: IDENTIFICAÇÃO DE PADRÕES	19
3.1.5	INTERFACE WEB DO UNBOX	20
3.2	CENÁRIO DE ANÁLISE: UNB	25
4	RESULTADOS	26
4.1	ESTUDO DE CASO 1	26
4.1.1	ANÁLISE ESTÁTICA.....	27
4.1.2	ANÁLISE DE REDE	29
4.1.3	ARQUIVOS BAIXADOS	30
4.1.4	RESULTADO DO ESTUDO DE CASO 1	32
4.2	ESTUDO DE CASO 2	33
4.2.1	ANÁLISE ESTÁTICA.....	34
4.2.2	ANÁLISE DE REDE	36
4.2.3	ARQUIVOS BAIXADOS	37
4.2.4	RESULTADO DO ESTUDO DE CASO 2	37
5	CONCLUSÃO	39
	REFERÊNCIAS BIBLIOGRÁFICAS	41

LISTA DE FIGURAS

2.1	Spams reportados por ano (CERT,2013)	4
2.2	Exemplo de Phishing	5
2.3	Quantidade de Incidentes por categoria (CAIS,2013)	6
2.4	Porcentagem de incidentes relacionados à <i>bots</i> (CAIS,2013)	7
2.5	Tentativas de fraudes reportadas em 2013 (CERT.br, 2013)	8
2.6	Análise de Malware (BORGES; GOMES; DUARTE, 2012)	9
2.7	Comando <i>md5sum</i>	10
2.8	Etapas de ofuscamento de executável (SBSeg, 2011)	13
3.1	Arquitetura de funcionamento do UnBox (Cuckoo Sandbox Book - Release 1.1)	16
3.2	Processo de Análise UnBox (BLACKHAT USA, 2013)	18
3.3	Exemplo de Regra Yara (yara Documentation Release 3.2.0)	20
3.4	Tela Inicial UnBox	21
3.5	Tela de Envio de Artefatos UnBox	21
3.6	Tela de Análises Recentes UnBox	22
3.7	Tela de Análises Pendentes UnBox	23
3.8	Tela de Busca UnBox	24
3.9	Página de relatórios do UnBox	25
4.1	Especificações do artefato	26
4.2	Capturas de tela e Hosts acessados	27
4.3	Arquivos acessados	27
4.4	Strings extraídas do artefato	28
4.5	algumas funções utilizadas pelo artefato	28
4.6	Consulta aos antivírus	29
4.7	IPs e Domínios acessados	30
4.8	Protocolo HTTP	30
4.9	Arquivos baixados	31
4.10	Chaves de Registro acessadas pelo arquivo <i>reviera[1].nil</i>	31
4.11	Análise de antivírus do arquivo <i>reviera[1].nil</i>	32
4.12	Detalhes do arquivo <i>chequedevolvido.cpl</i>	33
4.13	Arquivos acessados no Estudo de Caso 2	34
4.14	Chaves de registro	34
4.15	Bibliotecas acessadas na execução do artefato	35

4.16	Strings retiradas do artefato	35
4.17	Análise de alguns antivírus	36
4.18	IPs acessados	36
4.19	Protocolo HTTP	37
4.20	Arquivo baixado	37

LISTA DE TABELAS

2.1	DLLs mais utilizadas	12
-----	----------------------------	----

Lista de símbolos, nomenclatura e abreviações

CAIS - Centro de Atendimento a Incidentes de Segurança

CERT - Centro de Estudos, Resposta e Tratamento de Segurança no Brasil

DDoS - Distributed Denial of Service

DLL - Dynamic-link library

DNS - Domain Name System

DoS - Denial of Service

HTTP - Hypertext Transfer Protocol

IP - Internet Protocol

MBR - Master Boot Record

MD5 - Message Digest Algorithm 5

PCAP - packet capture

RNP - Rede Nacional de Ensino e Pesquisa

SHA-1 - American Standard Code for Information Interchange

SPAM - Sending and Posting Advertisement in Mass

SSH - Secure Shell

URL - Uniform Resource Location.

VNC - Virtual Network Computing

Capítulo 1

Introdução

No contexto atual de Redes de Comunicações altamente interligadas, diversos objetivos são traçados de modo a utilizar tais recursos. Como é utilizada para dar mais comodidade ao usuários, as Redes de Computadores são alvos de inúmeros ataques, de modo a roubar dados, infectar máquinas para utilizá-las em fins obscuros, causar transtornos na rede, entre outros. Tais violações podem ser alcançadas de diversas formas, como por invasão via arquivos maliciosos que atuam de diversas maneiras, que compreende desde o roubo de dados até o dowload de arquivos maliciosos.

Num contexto universitário, dados valiosos podem ser almeçados por *hackers*, como resultados de pesquisas, dados de alunos, professores e funcionários, dados administrativos da Universidade, entre outros. Tais dados, embora protegidos por um sólido esquema de segurança, podem se tornar vulneráveis através da ação de artefatos maliciosos que objetivam roubar os dados em si ou roubar os meios de acesso a essas informações. É importante, assim, poder observar o teor dos artefatos recebidos em toda a realidade acadêmica, de forma a poder observar a ação destes artefatos e, assim, compreender se sua ação se configura como maliciosa

1.1 Objetivos

Neste trabalho têm-se como objetivo analisar artefatos potencialmente maliciosos em uma ferramenta automatizada, online e personalizada para o uso universitário, de forma a obter informações precisas a respeito de diversos arquivos ou URLs suspeitas, visando o aumento da proteção de microcomputadores e outros dispositivos de usuários comuns, isto é, aqueles que utilizam a Internet de forma geral e que, normalmente, não possuem toda a gama de conhecimentos em Segurança de Redes. Com isso, busca-se proporcionar uma alternativa de utilização segura da Rede Mundial de Computadores, mediante a observação de artefatos desconhecidos.

1.2 Objetivos Específicos

Têm-se como objetivo específico os seguintes pontos:

- Implementação da ferramenta em uma rede real: Têm-se como objetivo implementar o UnBox, Framework de análise de artefatos maliciosos em uma rede real, que no caso deste trabalho é a rede de computadores da Universidade de Brasília (UnB). Com isso, busca-se disponibilizar a ferramenta à comunidade acadêmica, de forma a possibilitar a utilização por todos os componentes desta comunidade (professores, servidores e alunos);
- Analisar, via UnBox, artefatos maliciosos e observar suas características, comportamentos e ações tomadas a partir do momento de sua ativação. Toda a análise será realizada de forma automática e isolada, de forma a proporcionar segurança aos insumos que hospedam a ferramenta. Têm-se como objetivo a geração de relatórios ao fim das análises, de forma a observar todas as informações coletadas.
- Avaliar a ferramenta UnBox, bem como a técnica denominada *Sandbox*. Analisar pontos positivos, negativos, características principais e pontos a melhorar na ferramenta.

1.3 Justificativa

Com o advento das redes de computadores e da intensa utilização da Internet para as mais diversas tarefas, crimes cibernéticos têm sido amplamente usados para o roubo de informações, invasão de computadores particulares e monitoração de atividades *on-line*. Em grande parte dos casos, a invasão se dá por meio da Engenharia Social de *hackers*. Diversos artefatos são enviados via email, links de propagandas e outros. Diante deste cenário, foi observada a necessidade de uma solução de análise de artefatos suspeitos antes da execução dos arquivos, de forma a proporcionar um maior grau de segurança aos usuários da Internet.

No contexto da Universidade de Brasília, foi observado que há a necessidade de uma ferramenta capaz de analisar o teor de artefatos recebidos via email, pendrives ou outros meios, e que tais artefatos possam ser incorporados à rede universitária mesmo sendo maliciosos. Com isso busca-se fornecer um maior grau de segurança à realidade universitária no quesito Rede de Comunicação.

1.4 Organização do Trabalho

O trabalho é organizado de forma sequencial, de forma a permitir ao leitor a total compreensão dos objetivos do trabalho e técnicas utilizadas. O capítulo 2 apresenta os principais conceitos teóricos utilizados no trabalho, permitindo a compreensão das operações e técnicas utilizadas na implementação do trabalho. O capítulo 3 apresenta a ferramenta UnBox, Framework utilizado na análise de artefatos maliciosos, seu funcionamento, componentes e formas de operação, além dos procedimentos práticos adotados no trabalho, isto é, quais as ações tomadas com o objetivo da obtenção dos resultados práticos. O capítulo 4 apresenta os resultados obtidos através do Framework UnBox em dois estudos de caso. É mostrado os procedimentos realizados nas análises, quais as características de cada artefato analisado e é apresentada a melhor forma de classificá-los, com base na conceituação teórica apresentada. No capítulo 5 é apresentada as conclusões

obtidas neste trabalho, isto é, quais as impressões obtidas no decorrer da execução das atividades, quais os objetivos alcançados e propostas para trabalhos futuros a serem executados nesta linha de pesquisa.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão apresentados conceitos importantes utilizados na execução deste trabalho. Serão mostradas a seguir os principais tipos de ameaças presentes na internet, algumas classes de códigos maliciosos e informações sobre as técnicas de análise utilizadas pela ferramenta.

2.1 Spam e *Phishing*

Spam é o nome dado a mensagens recebidas sem terem sido solicitadas, na maioria das vezes propaganda. Normalmente o remete do SPAM (conhecido com *Spammer*) utiliza-se de softwares para coletar e-mails em sites e listas de discussão em que o usuário tenha se cadastrado, outro método utilizado é formação de endereços de e-mail através da combinação de uma lista de palavras.

A Figura 2.1 Mostra o número de Spams reportados ao CERT por ano.

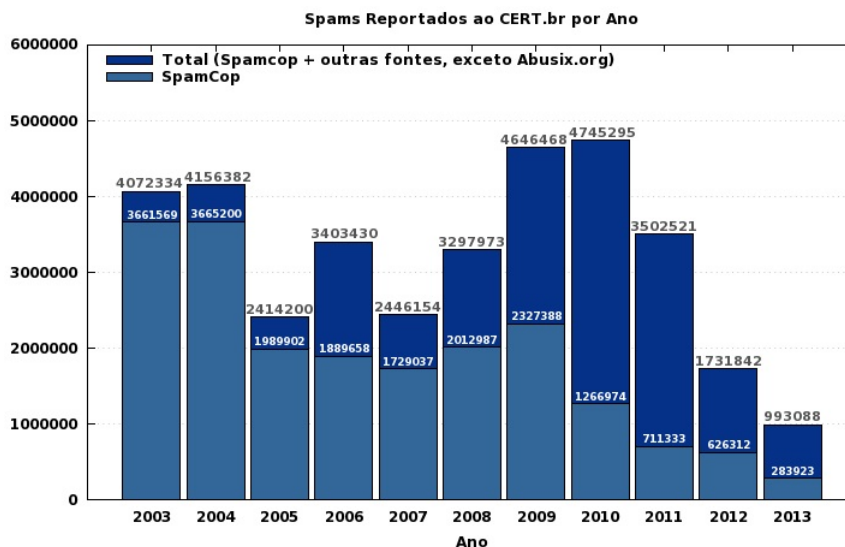


Figura 2.1: Spams reportados por ano (CERT,2013)

Já o termo *phishing* (vem de “*Fishing*”, que significa pescaria), refere-se ao meio de coleta de informações que emprega um e-mail enviado ao usuário como uma forma de “isca”. O atacante utiliza essas mensagens para coletar informações como: dados bancários, login e senha de e-mail.

A Figura 2.2 apresenta um exemplo de phishing, onde o atacante envia uma mensagem falsa de uma suposta investigação do ministério público federal, quando o usuário clica em um link é redirecionado para uma página falsa com um formulário para coleta de dados.



Figura 2.2: Exemplo de Phishing

2.2 Códigos Maliciosos

Códigos maliciosos ou *malwares* são programas desenvolvidos para executar atividades maliciosas em um computador, como por exemplo, roubo de informações e indisponibilização de serviços. *Malwares* podem ser classificados de acordo com o seu comportamento, porém é comum um código malicioso desempenhar mais de uma função podendo fazer parte de mais de uma classe de comportamento.

A Figura 2.3, do Centro de Atendimento a Incidentes de Segurança (CAIS), mostra que os códigos maliciosos representam a maior parte dos incidentes de segurança reportados na rede acadêmica da RNP (Rede Nacional de Ensino e Pesquisa) em que a Universidade de Brasília faz parte com cerca de 67130 incidentes.

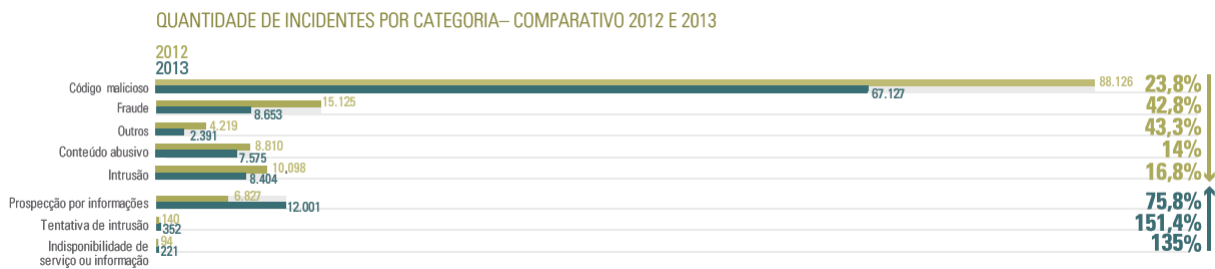


Figura 2.3: Quantidade de Incidentes por categoria (CAIS,2013)

Apresentaremos a seguir algumas classes de código malicioso.

2.2.1 Vírus

Um vírus é um código malicioso com capacidade de auto replicação, habilidade de criar cópias de si mesmo e distribuir para outros arquivos ou programas. Normalmente vírus necessitam da ação de um fator externo (por exemplo, a execução do programa infectado por um usuário) para executarem suas ações. Os vírus podem ser divididos em duas categorias, os vírus compilados e os interpretados(SBSEg, 2011):

- Vírus Compilados: Tipo de vírus que é compilado em alguma linguagem de programação para ser executado em determinado Sistema operacional. Vírus compilados podem infectar um programa, que se propaga para infectar outro programa ou podem infectar o setor MBR (*Master Boot Record*) de um disco ou mídia removível;
- Vírus Interpretados: O código fonte desse tipo de vírus só pode ser executado por uma aplicação específica e não por um sistema operacional, como os vírus compilados. Esse tipo de vírus pode ser escrito em linguagem de script ou de macro:
 - Vírus script utilizam linguagens como JavaScript e VBScript, podendo ser enviados por e-mail ou acessados através de páginas web;
 - Vírus de macro utilizam programas que utilizam macros, editores de texto e planilhas eletrônicas, por exemplo, para se propagarem. Se espalhando quando um documento infectado e com o recurso de macro habilitado é executado.

2.2.2 Worms

Código malicioso capaz de se propagar através da rede, sem que precisem, para isso, da execução por um agente externo, pois exploram vulnerabilidades existentes em softwares ou no próprio sistema.

Os *worms* consomem uma grande quantidade de recursos no computador ou na rede que foram contaminados, isso se deve a sua capacidade de propagação sem necessitar da ação de um usuário.

O primeiro *worm* conhecido foi criado por Robert Tappan Morris em 1988, recebeu o nome de *Morris Worm* e tinha como objetivo apenas realizar uma contagem dos computadores ligados a internet. O *Morris Worm* ao chegar numa máquina enviava uma mensagem para incrementar a contagem e identificava os computadores ligados a ela, porém antes de se propagar a outra máquina fazia uma checagem para verificar se o computador já estava com o *worm*, se já estivesse não seria novamente invadido, porém um erro no código fez com que esse mecanismo falhasse e o *Morris Worm* invadisse diversas vezes o mesmo computador causando diversos danos a empresas e universidades conectadas a internet na época.

2.2.3 Bot e botnet

Assim como os *worms*, as *bots* tem capacidade de se propagar automaticamente. Após a infecção o atacante se comunica remotamente com o computador infectado, conhecido com zumbi, podendo utiliza-lo para propagar malwares, enviar Spams ou realizar ataques de negação de serviço (DOS).

A *botnet* é uma rede formada por computadores zumbis de forma a realizar ataques coordenados aumentando a eficácia da ação realizada pela *bot*. Atacantes também podem oferecer serviços de “aluguel” de *botnet* para realização de ataques ou envio de spam.

Segundo o Centro de Atendimento a Incidentes de Segurança (CAIS) os incidentes relacionados à bots representaram em 2013 certa de 90% do total de incidentes reportados na rede acadêmica, entre as Bots destaca-se o Conficker com 40814 incidentes, conforme apresenta a Figura 2.4.

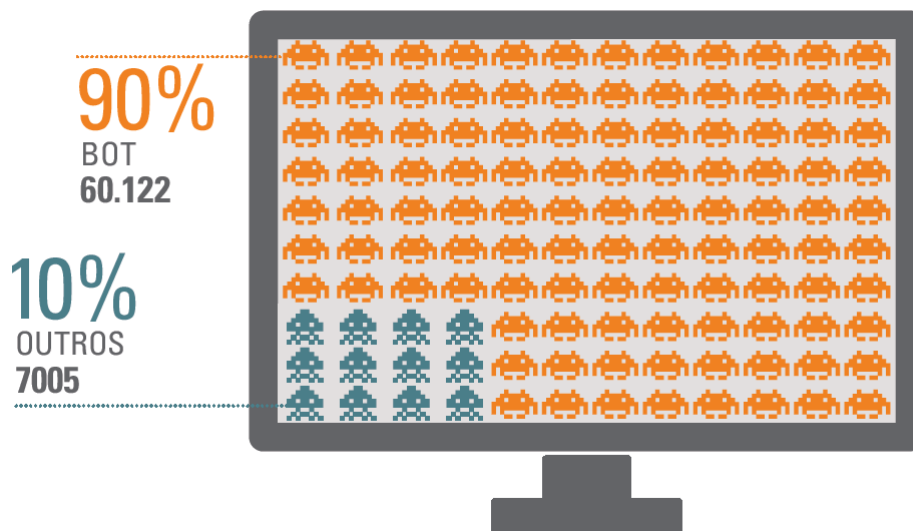


Figura 2.4: Porcentagem de incidentes relacionados à bots (CAIS,2013)

2.2.4 Cavalo de Tróia (*trojan*)

Esse tipo de código malicioso tenta enganar o usuário se passando por um programa que executa tarefas legítimas, ofuscando assim suas ações maliciosas.

Bastantes utilizados no cenário brasileiro para coletar informações para fraudes financeiras. As tentativas de fraudes utilizando cavalo de Tróia representam 27,40% do total das fraudes reportadas ao Centro de Estudos, Resposta e Tratamento de Segurança no Brasil (CERT), perdendo apenas para páginas falsas, conforme pode ser verificado na Figura 2.5.



Figura 2.5: Tentativas de fraudes reportadas em 2013 (CERT.br, 2013)

Cavalos de Tróia podem ser classificados de acordo com as suas ações, a seguir serão mostrados alguns tipos de *trojan* (CERT,2012):

- *Downloader*: baixa e instala outros códigos maliciosos;
- *Dropper*: instala outros códigos maliciosos obtidos junto ao próprio *trojan*;
- *Destrutivo*: altera ou apaga arquivos podendo até formatar o disco rígido deixando o sistema inoperante;
- *Banker*: Cavalo de Tróia que furta dados bancários do usuário

2.2.5 *Spyware*

Código Malicioso que monitora o sistema coletando informações e as enviando para terceiros. Alguns tipos de *spyware* são:

- *Keylogger*: captura as teclas digitadas no teclado, podem coletar, por exemplo, dados de acesso de serviços de e-mail ou de internet *banking* e códigos de cartão de crédito;
- *Mouseloggers*: *spyware* que coleta a posição do clique do mouse, podendo capturar informações de clicks em teclados virtuais utilizados por sistemas de bancos na web;
- *Screenlogger*: efetua cópias da imagem de tela, podendo capturar telas no momento do clique do mouse por exemplo. Com isso também pode coletar informações de teclados virtuais.

2.2.6 *Rootkit*

O atacante utiliza o *rootkit* com a intenção de esconder evidências da sua invasão, eliminando logs, por exemplo, e dessa forma mantendo o acesso de forma discreta ao sistema comprometido. *Rootkits* podem inclusive corrigir as vulnerabilidades que o atacante utilizou como forma de esconder evidências e principalmente evitando que outro atacante invada o sistema.

O termo *rootkit* tem origem nos sistemas Unix, porém também existem *rootkits* para sistemas Windows.

2.2.7 *Backdoor*

Malware utilizado para garantir o retorno do atacante ao computador invadido, evitando que o procedimento de invasão seja repetido para uma mesma máquina. *Backdoors* podem habilitar serviços ou ainda substituir programas de acesso remoto, como SSH, telnet e VNC, por versões alteradas adicionando vulnerabilidades que serão utilizadas pelo invasor a cada retorno.

2.3 Análise de Malware

A análise de um artefato visa compreender suas características, quais ações serão realizadas, como são executados no sistema operacional, identificando se é ou não um artefato malicioso.

Existem dois tipos de análise de artefatos maliciosos, a análise estática e a análise dinâmica, como mostra a Figura 2.6.

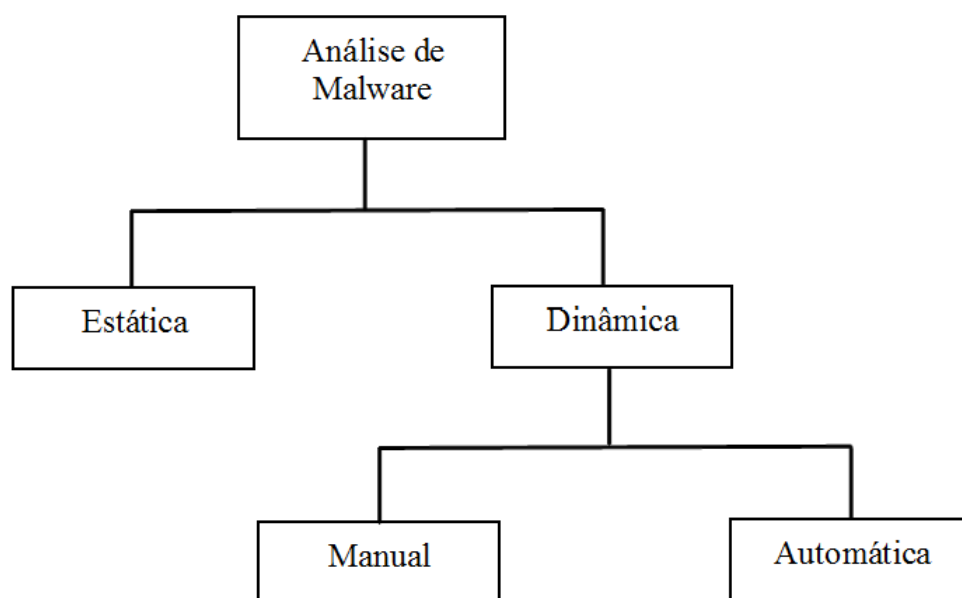


Figura 2.6: Análise de Malware (BORGES; GOMES; DUARTE, 2012)

Na análise estática o malware é analisado sem que ocorra a execução do artefato. Já na análise dinâmica ocorre a execução do malware, podendo-se avaliar o comportamento do código malicioso no sistema.

2.3.1 Análise Estática

Neste tipo de análise, são levantadas características do artefato malicioso sem que ocorra a execução. Alguns recursos e conceitos utilizados nesse tipo de análise serão apresentados a seguir.

2.3.1.1 Análise de Código

Pode-se analisar um código malicioso através de suas instruções. Para isso é necessário realizar a conversão de um arquivo binário em um código assembly ou outra linguagem de alto nível. As seguintes ferramentas podem ser utilizadas:

- Decompilador, ferramenta para traduzir um código binário em linguagem de alto nível. Um decompilador tenta identificar estruturas conhecidas no código binário as traduzindo em linguagem de programação (por exemplo C ou C++) o código gerado não é inteiramente igual ao código original;
- Disassembler, ferramenta semelhante ao decompilador porém traduz o binário em código assembly.

2.3.1.2 Função de Hash

É uma função unidirecional, onde uma entrada gera uma saída única de tamanho fixo e conhecendo-se a saída não é possível obter a entrada. Algumas funções de *hash* utilizadas são o *MD5*, o *SHA1* e o *SHA256*.

Um *hash md5* pode ser gerado através do comando "*md5sum*", presente de forma nativa na maior parte das distribuições linux. Na figura 2.7 temos um exemplo do uso do comando "*md5sum*" em que foi gerado um *hash* para o arquivo *bash*, tendo conhecimento do *hash* do arquivo original pode-se identificar se o arquivo foi alterado ou substituído.

```
root@unbox:~# md5sum /bin/bash
144968564a6b1159ed82059920cea76f /bin/bash
```

Figura 2.7: Comando *md5sum*

Uma função de *hash* pode ser usada para gerar uma identificação única de um *malware*. Desta forma pode-se realizar buscas e fazer comparações de artefatos de forma confiável e independente do nome do arquivo, possibilitando a procura e o compartilhamento de análises realizadas.

2.3.1.3 Análise de *Strings*

Pode-se identificar características de um código malicioso através de *strings*, cadeias de caracteres extraídas do artefato suspeito. Podem ser extraídas de um artefato cadeias de caracteres como por exemplo: mensagens de telas de interação com a vítima, mensagens de erros e URLs ou IPs que acusam endereços com o que o artefato pode tentar se comunicar. Um método utilizado para a análise de *strings* é a comparação com listas de palavras comumente encontradas em determinados tipos *malwares*.

2.3.1.4 Bibliotecas e Funções Utilizadas

O conhecimento de funções ou bibliotecas do sistema importadas por um executável pode trazer informações importantes sobre suas funcionalidades no sistema. As bibliotecas compartilhadas pelo Windows recebem o nome de *Dinamic Link Library* (DLL), na tabela 2.1 temos a lista das DLLs mais comuns.

Tabela 2.1: DLLs mais utilizadas

DLL	Descrição
<i>Kernel32.dll</i>	È uma DLL muito comum que contém funcionalidades principais, como acesso e manipulação de memória, arquivos e <i>hardware</i> .
<i>Advapi.dll</i>	Essa DLL provê acesso a componentes principais mais avançados do <i>Windows</i> , como o Gerenciador de Serviços e o Registro.
<i>User32.dll</i>	Essa DLL contém todos os componentes de interface de usuário como botões, barras de rolagem, e componentes de controle que respondem a ações de usuários.
<i>Gdi32.dll</i>	Essa DLL contém funções para apresentação e manipulação gráfica.
<i>Ntdll.dll</i>	Essa DLL é a interface para o <i>kernel</i> do <i>Windows</i> . Os executáveis normalmente não importam esse arquivo diretamente, mas ele é sempre importado indiretamente pelo <i>Kernel32.dll</i> . Se um executável importar esse arquivo, significa que o desenvolvedor pretende utilizar funcionalidades que normalmente não estão disponíveis nos programas do <i>Windows</i> .
<i>Wsock32.dll</i> e <i>Ws_32.dll</i>	Essas são DLLs de rede. Um programa que acessa qualquer uma delas na maioria das vezes se conecta a uma rede ou realiza tarefas relacionadas a rede.
<i>Wininet.dll</i>	Essa DLL contém funções de rede de nível alto, que implementam protocolos como FTP, HTTP e NTP.

2.3.1.5 Ofuscamento do Código

O desenvolvedor do malware pode utilizar técnicas para ofuscar um código malicioso dificultando sua identificação.

O ofuscamento do código pode ser alcançado por meio de uma compactação do executável, através de uma ferramenta conhecida como *packer*. Porém um malware compactado não executa suas funcionalidades sendo necessário que uma rotina de descompressão seja colocada junto ao código malicioso.

Outra maneira de se ofuscar um código é através do uso de criptografia, onde o executável é criptografado através de um *cryptor*. De maneira semelhante ao caso da compactação uma rotina para decifrar o executável é necessária para execução do código malicioso. A figura 2.8 apresenta as etapas utilizadas no ofuscamento de um executável.

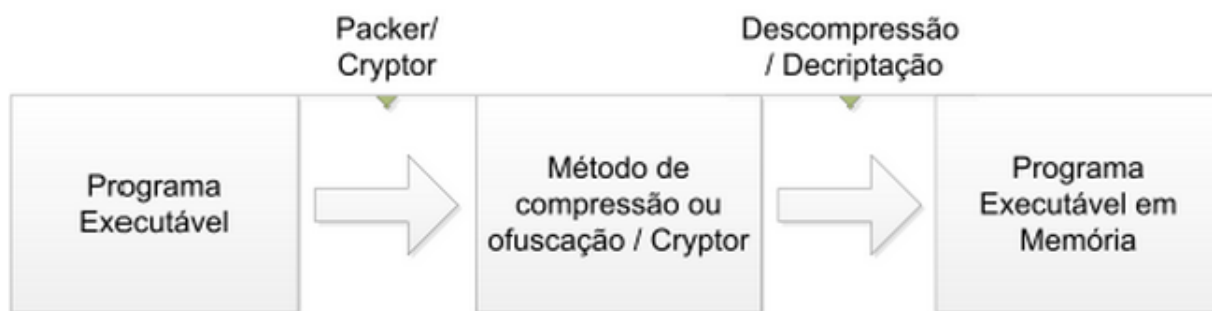


Figura 2.8: Etapas de ofuscamento de executável (SBSeg, 2011)

2.3.1.6 Antivírus

O antivírus é o recurso mais difundido de combate aos malwares. Um modo utilizado pelos antivírus para detecção de códigos maliciosos consiste em dividir o artefato suspeito em pequenos pedaços de códigos que são comparados com uma base de assinaturas, quando são encontradas similaridades o código é considerado malicioso.

Porém, com o crescimento no número de malwares, as atualizações das assinaturas que antes eram semanais, agora são feitas diariamente ou até mais de uma vez ao dia. Como essas assinaturas normalmente são geradas se forma manual, existe a dificuldade de gerar assinaturas de forma rápida.

O tempo médio de detecção de um malware após seu lançamento por um antivírus é de no mínimo 3 dias (PEOTTA,2012).

Outra forma de trabalho de um antivírus é através da execução em um pequeno emulador para identificar ações suspeitas, esta forma de identificação se encaixa no conceito de análise dinâmica, quem será tratado a seguir.

2.3.2 Análise Dinâmica

A análise dinâmica ocorre no tempo de execução do artefato suspeito, podendo listar todas as ações feitas pelo artefato malicioso como: bibliotecas acessadas, arquivos acessados, arquivos baixados e informações enviadas.

A análise dinâmica pode ser feita de forma manual ou automática, como pode ser observado na Figura 2.6. Na análise manual utiliza-se um depurador (*debugger*) que permite a execução e o monitoramento de um artefato em ambiente controlado (Borges,2012). O depurador pode executar artefato instrução após instrução acessando valores como: registradores, memória e conteúdo de pilha a cada instante.

Na análise dinâmica e automática o artefato é executado em um ambiente chamado *sandbox* que no contexto de análise de artefatos maliciosos, é um ambiente isolado e controlado utilizado para execução e monitoramento do comportamento de um artefato. Um analisador baseado em

sandbox executa o artefato malicioso nesse ambiente gerando um relatório com as ações executadas. O relatório deve ser analisado pelo usuário, para identificação do artefato.

Porém esse tipo de análise também possui limitações, como o tempo de execução de um *sandbox* é limitado a alguns minutos, pode-se deixar passar ações de malwares que só seriam executadas após certo tempo ou apenas após a reinicialização do sistema. Existem *malwares* que dependem de argumentos para serem executados, impossibilitando o monitoramento de suas ações por meio de um *sandbox*, já que esses argumentos não seriam passados.

Existem ainda códigos maliciosos feitos para não executarem em ambientes virtualizados, ou ainda feitos para funcionar em versões de sistemas operacionais diferentes daquela utilizada pelo *sandbox*, dificultando a análise nesses casos.

O anubis (<https://anubis.iseclab.org/>) e o malwr (<https://malwr.com/>) são exemplos de analisadores sandbox disponíveis para acesso gratuito.

2.3.2.1 *Cuckoo Sandbox*

O *Cuckoo Sandbox* é um projeto *open source* de analisador de malware, foi desenvolvido inicialmente por Claudio Guarnieri durante o *Google Summer of Code 2010*, projeto do google que oferece incentivos a desenvolvedores de ferramentas *open source*. Claudio Guarnieri ainda é o principal desenvolvedor e trabalha em conjunto com colaboradores que aderiram ao projeto.

O Cuckoo pode apresentar os seguintes resultados após sua análise:

- Informações sobre API acessadas e processos gerados pelo artefato suspeito;
- Captura do histórico de navegação do artefato;
- Arquivos acessados durante a execução;
- Capturas de tela durante a execução do artefato suspeito;
- Dump de memória da máquina virtual durante a execução;
- *Strings* encontrados no artefato.

Além da análise através de *sandbox* o *Cuckoo* realiza técnicas de análise estática como: análises de *strings* e de bibliotecas e funções importadas e a reputação em diversos antivírus.

O projeto *Malwr (Malware Analysis by Cuckoo Sandbox)* é um exemplo de analisador de *malware* implementado com base no *Cuckoo Sandbox* e disponibilizado para acesso gratuito.

Capítulo 3

Desenvolvimento

3.1 UnBox: Framework de Análise

Para a análise de códigos maliciosos, foi utilizada a técnica de *Sandbox*, que consiste em executar um arquivo desconhecido ou não confiável e de fontes desconhecidas, de forma a analisar o comportamento e definir o tipo de ação que o arquivo toma em execução.

O Framework utilizado na análise de artefatos maliciosos em ambiente de produção foi o UnBox, que é uma ferramenta automatizada de análise de códigos malicioso baseada no *Cuckoo Sandbox*, software livre e de código aberto e desenvolvido para análises de códigos maliciosos. O UnBox é, em resumo, uma customização do *Cuckoo*, sendo adequado para o uso em redes acadêmicas, mais especificamente na Universidade de Brasília, pois o UnBox foi customizado em *Framework para Análise Dinâmica de Códigos Maliciosos (ABREU, R. N. P.; CIDADE, T. F. V. (2013))*. O framework trabalha da seguinte forma: é disponibilizada uma página web ao usuário onde é possível fazer o upload de um código suspeito, ou ainda passar como informação uma URL não confiável. A análise é realizada de forma automática, pois o UnBox conta com uma arquitetura de máquina virtual, onde é simulada, em ambiente controlado, a ação do arquivo suspeito ou URL insegura. Em seguida, é repassado ao usuário um relatório gerado pelo UnBox das principais ações do arquivo ou URL, sendo possível por parte do usuário analisar e concluir a respeito da periculosidade do artefato.

É possível analisar diversos tipos de arquivos com o UnBox, tais como:

- Executáveis genéricos do Windows;
- Arquivos DLL;
- Arquivos PDF;
- Documentos *Microsoft Office* (Word, Excel, PowerPoint e outros);
- URLs e arquivos HTML;
- Scripts PHP;

- Scripts Visual Basic (VB);
- Arquivos ZIP;
- Arquivos JAR.

É importante ressaltar que a análise de tipos de arquivos citados acima depende de uma customização adequada da ferramenta, bem como dos insumos utilizados pela ferramenta, tais como máquina virtual, rede virtual e outros.

3.1.1 Funcionamento do UnBox

O UnBox possui uma arquitetura que proporciona análises de códigos maliciosos de forma segura e completa. A figura abaixo apresenta o modelo de funcionamento do UnBox, bem como os componentes necessários para a implementação da ferramenta.

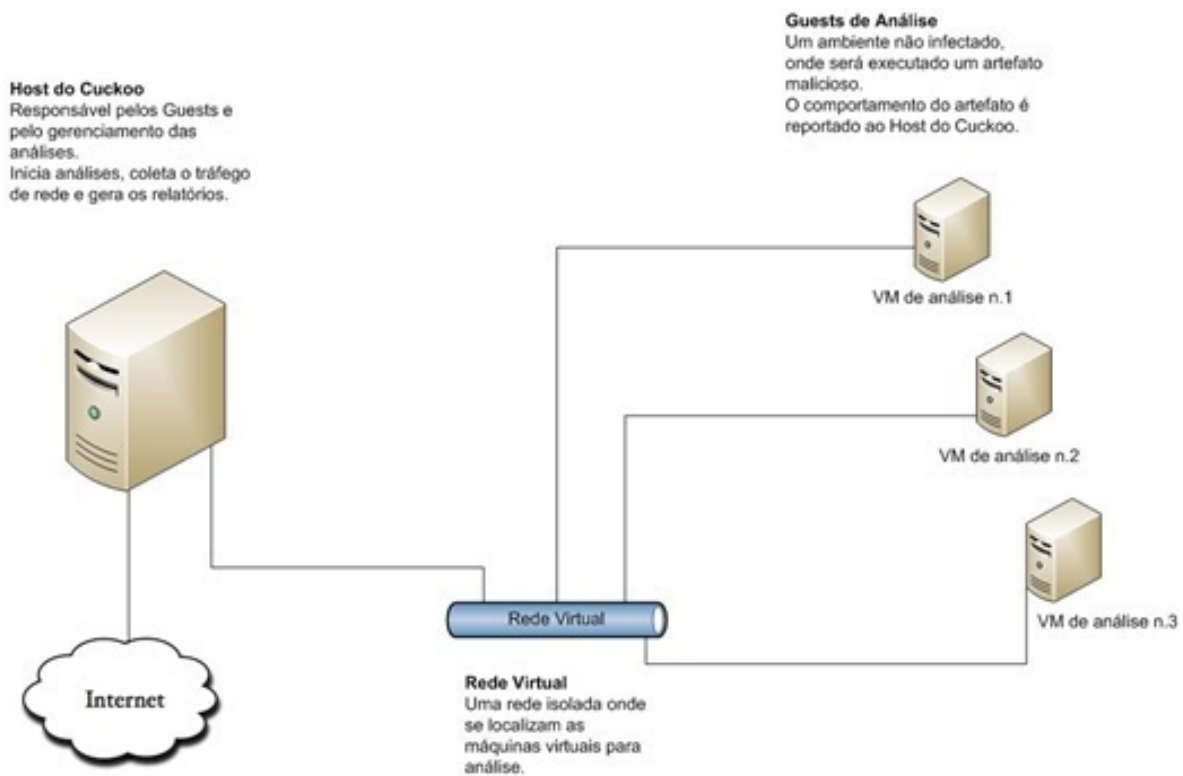


Figura 3.1: Arquitetura de funcionamento do UnBox (Cuckoo Sandbox Book - Release 1.1)

Pode-se observar os elementos que compõem a arquitetura de análise do UnBox. É ressaltado aqui suas principais características, bem como suas funções no cenário de análise.

- Host Cuckoo: É responsável por boa parte da análise do artefato malicioso. Aqui é configurada a aplicação UnBox, de forma a apresentar ao usuário uma interface *web* para o repasse

do artefato ou URL suspeita. Aqui estão presentes também os arquivos de configuração do UnBox, tais como, arquivo de configuração de Rede virtual (tratada mais adiante), arquivos de configuração de interface *web*, entre outros. É importante ressaltar que o host é o único elemento que possui interface de saída para a Internet, aqui é captado o tráfego gerado pelo artefato durante a análise, sendo responsável, assim, por registrar tais atividades para a geração de relatórios.

- Rede Virtual: Este elemento é criado entre o Host Cuckoo e os ambientes virtualizados que realizarão a análise dos códigos submetidos. De acordo com a sugestão indicada pela documentação da ferramenta Cuckoo, o virtualizador utilizado foi o *VirtualBox*, de forma a criar um ambiente virtualizado Windows para os testes de arquivos e URLs submetidos ao UnBox. O *VirtualBox* cria uma rede virtual com base em um range de IPs privados. No caso deste trabalho, a rede utilizada foi 192.168.56.0/24, onde o Host Cuckoo recebeu o IP 192.168.56.1 e o ambiente virtualizado recebeu o IP 192.168.56.101. É importante ressaltar que a rede virtual trabalha de forma isolada, de forma que não há risco de o artefato malicioso realizar atividades no Host Cuckoo ou no ambiente de trabalho do usuário, configurando segurança para a análise do artefato e também para o usuário do sistema.
- Guest de Análise: Neste elemento é realizada a análise do artefato ou URL submetida. Este ambiente, que é totalmente virtual, deve ser preparado para o tipo de análise específica ou o tipo de ambiente que se deseja simular. Este ambiente é configurado de forma a estar sempre preparado para receber do Host o artefato a ser analisado, e simula o funcionamento deste artefato, salvando imagens da ação do código, que são mostradas na interface *web*, ao final da análise. Ao final de cada análise, é restaurado um *Snapshot* da máquina inicial, de forma que a cada nova análise, não sejam encontrados vestígios da análise anteriormente executada.

3.1.2 Etapas de Análise do UnBox

É importante entender o processo de funcionamento do UnBox, isto é, quais são as etapas que compõem a análise de um artefato malicioso ou URL suspeita. A figura 3.2 mostra um esquema simplificado, que será explicado a seguir.

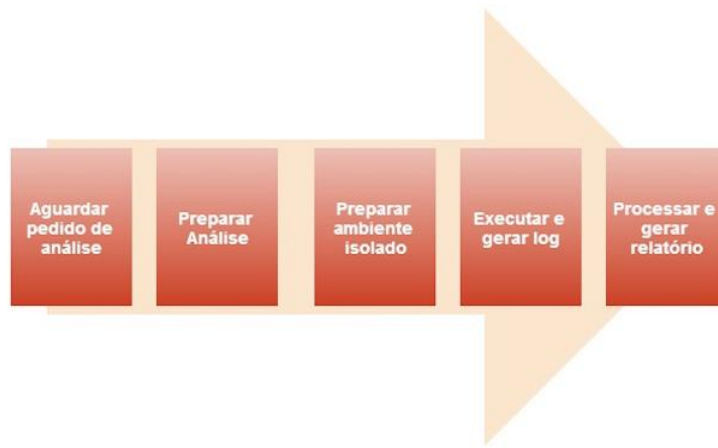


Figura 3.2: Processo de Análise UnBox (BLACKHAT USA, 2013)

De acordo com a figura 3.2, é possível identificar os principais componentes da análise realizada pelo UnBox:

- **Aguardar Pedido de Análise:** Aqui o processo do UnBox em execução no *host* aguarda o pedido de análise, que acontecerá via interface *web*.
- **Preparar Análise:** Após a requisição de análise realizada via interface *web* por parte do usuário, os arquivos de configuração do UnBox são acionados e o ambiente virtualizado é ativado, para que seja iniciada a análise.
- **Preparar Ambiente Isolado:** Aqui o artefato é repassado para o ambiente virtual, e o artefato é inicializado, de forma a observar seu comportamento e ações a serem tomadas pelo código.
- **Executar e Gerar Log:** Aqui o código malicioso está em execução, e todas as atividades estão sendo avaliadas e registradas, de forma a gerar insumos para a confecção do relatório. Estas informações são enviadas ao Host, que é responsável por mostrar os resultados ao usuário.
- **Processar e Gerar Relatório:** Nesta etapa a execução foi finalizada e o relatório completo de análise é mostrado ao usuário. Diversas informações são expostas ao usuário, de forma a informar a ação tomada durante a análise do *malware*.

3.1.3 Módulos do UnBox

O UnBox é composto por diversos módulos, que são responsáveis pela separação na configuração da ferramenta, isto é, cada módulo é responsável por uma parte específica na arquitetura de funcionamento do UnBox, de forma que a configuração e customizações do Framework se tornam mais simples e fáceis de serem organizadas.

Os módulos basicamente definem as principais formas de interação entre os elementos da arquitetura. Os principais módulos, que são definidos por arquivos de configuração no UnBox, são:

- *Auxiliary Module*: Módulo que define parâmetros auxiliares da análise de artefatos. Este módulo define atividades como: captura de pacotes gerados durante a análise, utilizando a ferramenta *Tcpdump*, e interface de rede a ser utilizada na atividade de ação do artefato, que no caso padrão é a interface *vboxnet0*, uma interface de rede virtual.
- *Machinery Module*: Este módulo define os principais parâmetros de interação entre o Host e o sistema de virtualização. Por padrão, este módulo é configurado para interação com o sistema *VirtualBox*, que é previamente definido no arquivo *cuckoo.conf*, entretanto é possível alterar esta opção, configurando este módulo. Neste módulo é definido também os parâmetros da máquina virtual utilizada para a análise dos artefatos maliciosos.
- *Processing Module*: Este módulo é responsável pelas informações coletadas durante a análise, bem como os tipos de dados a serem repassados durante e após a análise. Aqui é selecionado os dados para a criação de relatórios.
- *Analysers Module*: Módulo responsável pela definição de interações entre o sistema analisador e a execução do artefato malicioso. Podem ser definidos diversos módulos analisadores, que são responsáveis por atividades específicas. Aqui também é feita a captura de telas da máquina virtual, que serão utilizadas para informar ao usuário as ações tomadas pelo artefato. É responsável ainda pela simulação de interações humanas, como clique de mouse, confirmação via botões, e outros.

3.1.4 Yara: Identificação de padrões

O *Yara* é uma ferramenta utilizada para criar regras e padrões de códigos maliciosos com base em *Strings* - cadeias de caracteres. É possível criar descrições de famílias de *malwares* com base em textos ou padrões binários e cada regra de padrão consiste em um conjunto de *Strings* e expressões do tipo *boolean*, que determinam sua lógica.

A figura 3.3 ilustra um tipo de regra criada no *Yara*:

É possível observar que a criação de regras assemelha-se a tipos de arquivos criados em C. A regra baseia-se na definição de seu nome, declarado na primeira linha, além de parâmetros que a definem, como grau de periculosidade e descrição. Em seguida, são inseridas cadeias de caracteres que determinam o tipo de características presente em códigos maliciosos, que serão procuradas no artefato no momento de análise. Durante a análise, caso sejam encontradas tais cadeias de caracteres no artefato analisado, este é classificado como um artefato malicioso.

Com o *Yara*, é possível criar regras específicas para os mais diferentes usos e ambientes. Podem ser criadas regras para ambientes bancários a partir da inserção de palavras comum em ambientes de bancos e instituições. Podem também ser criadas regras para ambientes acadêmicos, com expressões comum em ambientes universitários, além de outros ambientes. Com isso, constata-se a força da ferramenta, que permite a configuração em diversos ambientes e em diversas situações de trabalho. Para este trabalho foram criadas regras que configuravam ações humanas, como clique de mouse em determinadas situações criadas pelo artefato, de forma a alcançar maior similaridade


```

rule silent_banker : banker
{
  meta:
    description = "This is just an example"
    thread_level = 3
    in_the_wild = true

  strings:
    $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
    $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
    $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

  condition:
    $a or $b or $c
}

```

Figura 3.3: Exemplo de Regra Yara (yara Documentation Release 3.2.0)

com ambientes reais em que artefatos maliciosos são configurados para executar suas ações.

3.1.5 Interface Web do UnBox

Como citado anteriormente, o UnBox é um framework baseado na ferramenta Cuckoo, que é uma ferramenta de código aberto e gratuita. Assim, o UnBox é uma customização do Cuckoo, criado para ser utilizado em um ambiente acadêmico, especificamente na Universidade de Brasília (UnB). Sua customização foi pensada para o contexto universitário, e as principais mudanças consistem na interface, que recebeu a imagem da Universidade como símbolo principal e na linguagem, que é o português.

A interface web do UnBox é bastante simples e amigável ao contato do usuário, possuindo uma tela inicial que mostra um resumo das atividades da ferramenta, como é mostrado na figura 3.4

Na figura 3.4 é possível observar o número de arquivos enviados para análise, bem como estatísticas a respeito da quantidade de arquivos em cada status de análise. Aqui o mais comum é observar análises do tipo *reported*, que são análises finalizadas e com resultados e relatórios disponíveis.

A seguir, na figura 3.5 é mostrada a tela de envio de artefatos maliciosos:

É possível observar as opções de seleção entre arquivos, que são escolhidos dentre os arquivos do PC do usuário, ou URLs, que podem ser repassadas com as ações Copiar e Colar. Na opção "Avançado", pode-se definir configurações como tipo de arquivo a ser analisado, *Timeout*, prioridade de análise, entre outros parâmetros que podem configurar tipos especiais de análise.

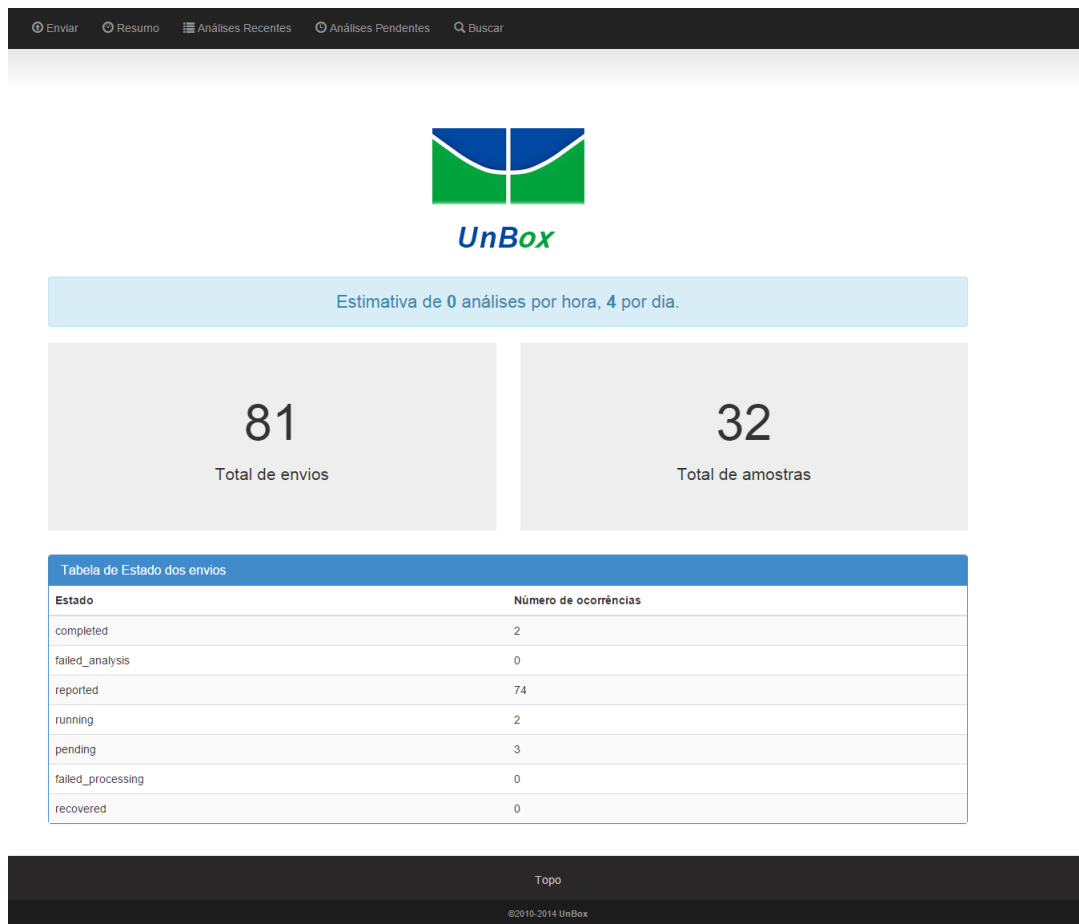


Figura 3.4: Tela Inicial UnBox

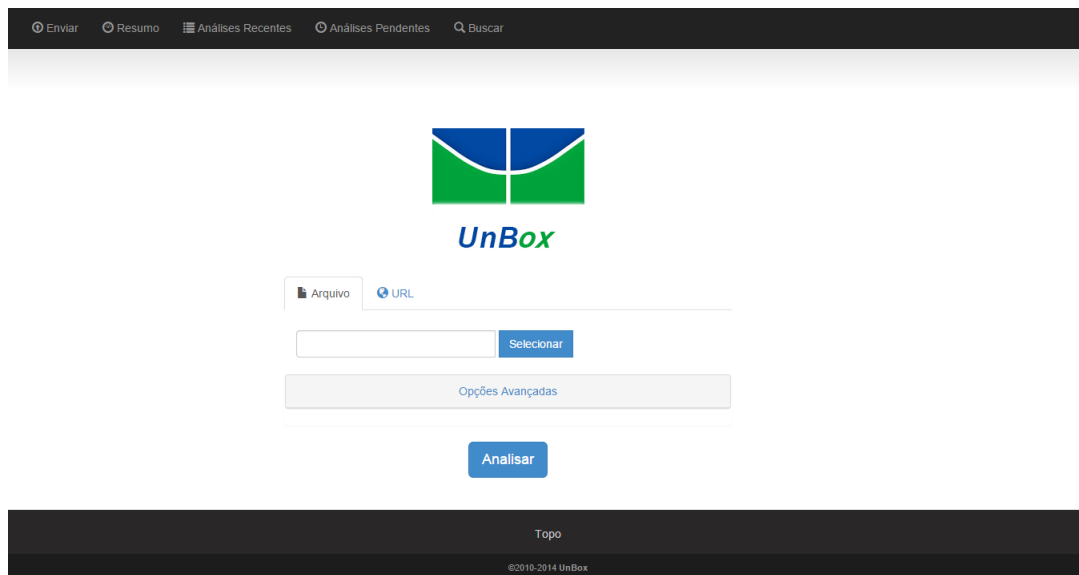
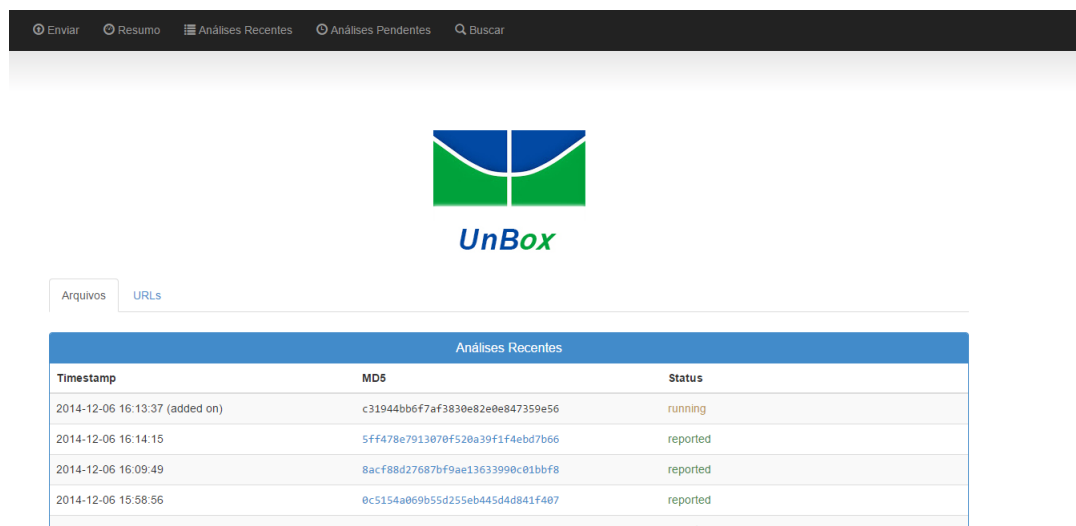


Figura 3.5: Tela de Envio de Artefatos UnBox

Em seguida, na figura 3.6 é exposta a tela de análises recentes:



Análises Recentes		
Timestamp	MD5	Status
2014-12-06 16:13:37 (added on)	c31944bb6f7af3830e82e0e847359e56	running
2014-12-06 16:14:15	5ff478e7913070f520a39f1f4ebd7b66	reported
2014-12-06 16:09:49	8acf88d27687bf9ae13633990c01bbf8	reported
2014-12-06 15:58:56	0c5154a069b55d255eb445d4d841f407	reported
2014-12-06 15:56:24	0c5154a069b55d255eb445d4d841f407	reported


Figura 3.6: Tela de Análises Recentes UnBox

Pode-se observar uma tabela com as últimas análises realizadas em arquivos, sendo possível mudar para análises de URLs analisadas. Com um clique simples é possível ver os detalhes de uma análise específica, sendo mostrados os principais dados desta análise, *Screenshots* da máquina virtual que realizou a análise e outros dados importantes da análise. É importante ressaltar que o UnBox mostra quando há uma análise em execução, através da marcação *added on*, que, dependendo do número de máquinas virtuais, pode contribuir para a criação de uma fila para análise.

Na figura 3.7 é mostrada a tela de Análises Pendentes do UnBox:

É possível observar a fila de arquivos a ser analisado pelo UnBox. É interessante ressaltar que o tipo do arquivo é mostrado, sendo possível observar se é um arquivo ou uma URL. Com isso é possível estimar o tempo de análise e a consequente redução da fila.

Enviar Resumo Análises Recentes Análises Pendentes Buscar



Fila para análise			
Timestamp	Category	Target	Status
2014-12-06 16:31:39 (added on)	file	/tmp/cuckoo-tmp/upload_5J7cIN/Verdade_AecioNeves 1994-2006.cpl	pending
2014-12-06 16:31:19 (added on)	file	/tmp/cuckoo-tmp/upload_FbdKOw/notafiscal.cpl	pending
2014-12-06 16:31:00 (added on)	file	/tmp/cuckoo-tmp/upload_WcQwqd/ComentarioDeVoz.exe	pending

Topo

©2010-2014 UnBox

Figura 3.7: Tela de Análises Pendentes UnBox

Por fim, na figura 3.8 é mostrada a tela de busca de análises do UnBox:

Aqui é possível observar que o UnBox possui diversos prefixos para facilitar a busca na ferramenta. Com essa opção, é possível recuperar análises antigas e manter o controle das análises, bem como recuperar análises com base em poucas informações, pois estas informações podem ser reforçadas com o uso de prefixos.

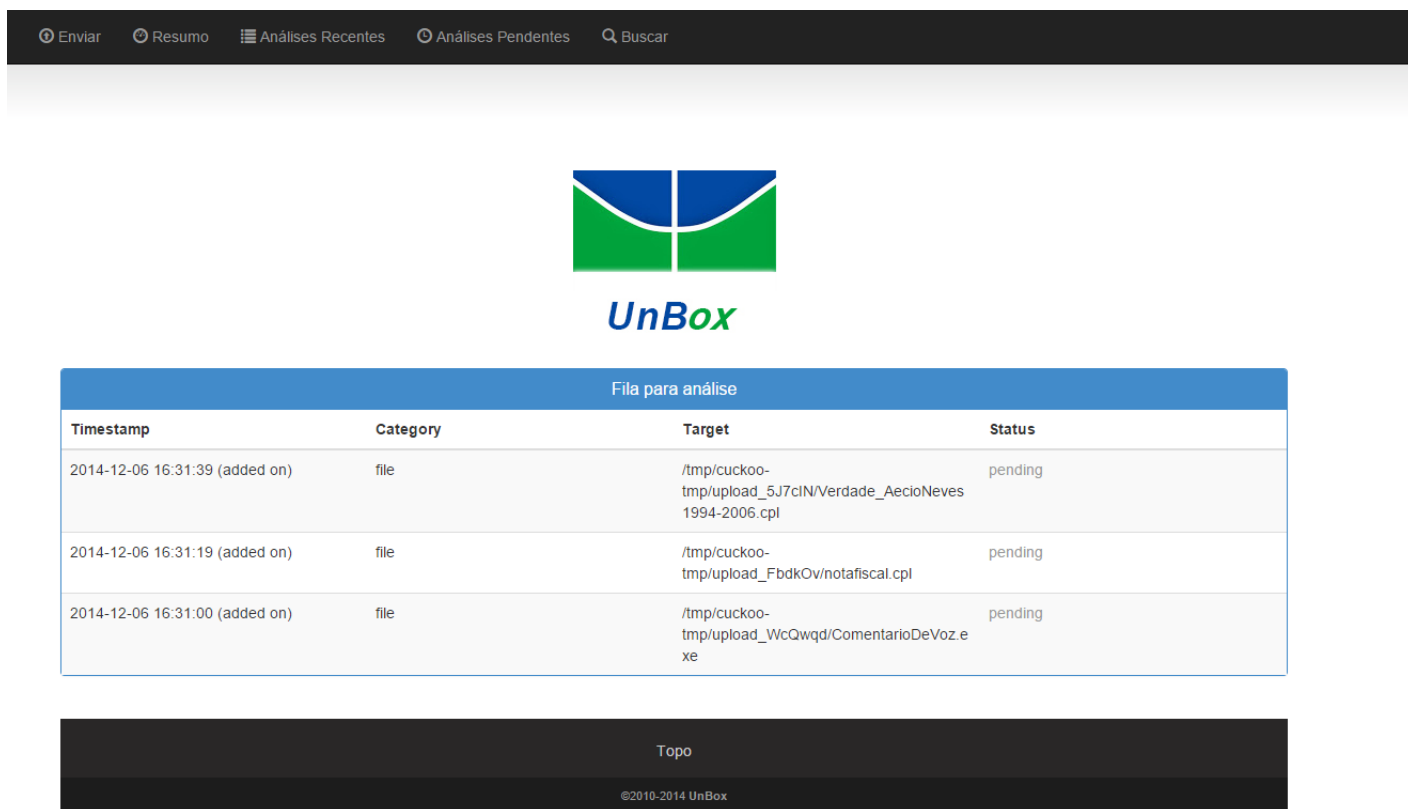


Figura 3.8: Tela de Busca UnBox

A página de relatórios do UnBox é dividida nas seguintes abas, conforme mostra a figura 3.9:

- **Resumo:** contém informações que caracterizam o artefato como: tamanho do arquivo, hashes criptográficos, regras do Yara acionadas, arquivos e chaves de registro abertos durante a execução;
- **Análise estática:** apresenta resultados da análise estática feita pelo UnBox como as bibliotecas importadas, strings extraídas e análise por antivírus;
- **Análise comportamental:** mostra detalhes dos processos gerados durante a análise;
- **Análise de rede:** mostra IPs, domínios e detalhes de protocolos de rede acessados durante a análise. Nesta aba é possível baixar toda a captura de rede gerada na execução do artefato em formato PCAP;
- **Arquivos baixados:** mostra todos os arquivos baixados durante a análise, nesta aba também é possível fazer o download desses arquivos.



Resumo	Análise Estática	Análise Comportamental	Análise de Rede	Arquivos Baixados
Análise				
Categoria	Início	Terminada	Duração	
FILE	2014-12-06 16:11:32	2014-12-06 16:14:15	163 seconds	

Figura 3.9: Página de relatórios do UnBox

3.2 Cenário de Análise: UnB

Para o desenvolvimento deste trabalho, o objetivo é implementar o framework UnBox em uma rede de produção e que a ferramenta possa ser utilizada em um contexto real de topologia de rede. Nesse cenário, a rede escolhida para a implementação foi a rede da Universidade de Brasília (UnB). Tal escolha se deve a alguns fatos:

- A vontade de deixar algo que seja utilizável pela Universidade e pela comunidade acadêmica (professores, servidores e alunos.)
- O fato de se observar o desempenho da ferramenta em um contexto real.
- A possibilidade de se tornar a ferramenta aberta a usuários sem o conhecimento técnico específico.

O UnBox exige, para um funcionamento mais rápido, de uma boa quantidade de memória, devido ao uso de máquinas virtuais na análise e de uma boa capacidade de processamento. Com isso a ferramenta foi instalada em um ambiente que tem disponível uma boa base de processamento e memória, sendo possível o pleno funcionamento da ferramenta.

A ferramenta encontra-se disponível atualmente em <http://unbox.unb.br:8080> e pode ser utilizada por quaisquer usuário que deseja realizar uma análise de artefato malicioso.

Capítulo 4

Resultados

4.1 Estudo de Caso 1

Foi baixado o artefato hospedado em <http://mosbeck.com.br/webalizer/Receita%20Federal.rar> no dia 05/12/2014, foi extraído o arquivo com o nome “*Receita Federal.cpl*” e analisado no UnBox.

Na aba resumo do UnBox podemos identificar algumas características do artefato conforme mostra a Figura 4.1.

The screenshot shows the UnBox interface with a navigation bar at the top containing 'Resumo', 'Análise Estática', 'Análise Comportamental', 'Análise de Rede', and 'Arquivos Baixados'. Below this is a table titled 'Análise' with the following data:

Categoria	Início	Terminada	Duração
FILE	2014-12-05 15:39:04	2014-12-05 15:41:42	158 seconds

Below the table is a section titled 'Especificações do artefato' with the following details:

Nome	Receita Federal.cpl
Tamanho	478720 bytes
Tipo	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
MD5	52c5baec841dc2790261765c75fe3f07
SHA1	16179c9ec48c151067025e9a5a6ee65545c202ff
SHA256	b406bf6224c0fb0a9ae632049a36a0e61ee09b81abc1c9df06fd39e1e86b3d03
SHA512	129eb157dc0c8a460c560e7e478ec1a1085bf30a9dd256e91d8f371ad04c14911888d3ea5d13ddd7c2f6e548edf7fcc9695a48b4d116f52f7c97267db3a95f29
CRC32	02D540B2
Ssdeep	12288:oR9eyFYK/IGRgOUqmq9kR6lhKXbqR75kyPEalEBkM:GeyfYK/cRgOnmq9g6lv78IEBkM
Yara	Não encontrado

Figura 4.1: Especificações do artefato

A análise durou 158 segundos e o artefato foi identificado como um executável do Windows. Temos ainda informações como o tamanho do arquivo, os *hashs* que identificam o arquivo de forma única (MD5, SHA1, SHA256, SHA512, CRC32). O *ssdeep* é uma função de *fuzzy hash*, com ela é possível identificar similaridades entre artefatos.

Nenhuma regra do Yara foi identificada no artefato, o Yara faz a análise das strings encontradas no arquivo suspeito.

Na figura 4.2 temos as capturas de tela, porém nenhuma informação pode ser tirada, pois o artefato é executado totalmente em segundo plano, e a lista de IP's e domínios acessados.



Figura 4.2: Capturas de tela e Hosts acessados

A figura 4.3 traz as lista dos arquivos acessados durante a execução do malware. Analisaremos alguns desses arquivos posteriormente.

```
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Temporary Internet Files
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Temporary Internet Files\Content.IE5
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Hist\xc3\xb3rico
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Hist\xc3\xb3rico\History.IE5
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Temporary Internet Files\Content.IE5\
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Temporary Internet Files\Content.IE5\index.dat
C:\Documents and Settings\cleosvaldo\Cookies\
C:\Documents and Settings\cleosvaldo\Cookies\index.dat
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Hist\xc3\xb3rico\History.IE5\
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Hist\xc3\xb3rico\History.IE5\index.dat
PIPE\lsarpc
C:\Documents and Settings\cleosvaldo\IETldCache\
C:\Documents and Settings\cleosvaldo\IETldCache\index.dat
PIPE\ROUTER
c:\autoexec.bat
C:\Documents and Settings
C:\Documents and Settings\cleosvaldo
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais
C:\Documents and Settings\All Users\Dados de aplicativos\Microsoft\Network\Connections\Pbk\*.pbk
C:\WINDOWS\system32\Ras\*.pbk
C:\Documents and Settings\cleosvaldo\Dados de aplicativos\Microsoft\Network\Connections\Pbk\*.pbk
C:\Documents and Settings\cleosvaldo\Configura\xc3\xa7\xc3\xb5es locais\Temporary Internet Files\Content.IE5\F53XN3FC\reviera[1].nil
C:\Documents and Settings\cleosvaldo\Dados de aplicativos\cleosvaldo.com
```

Figura 4.3: Arquivos acessados

4.1.1 Análise estática

Na aba “Análise estática” têm-se informações das strings extraídas do artefato, como mostra a figura 4.4, essas strings podem indicar o comportamento do malware e são usadas pelo Yara para identificação de padrões encontrados em malwares.

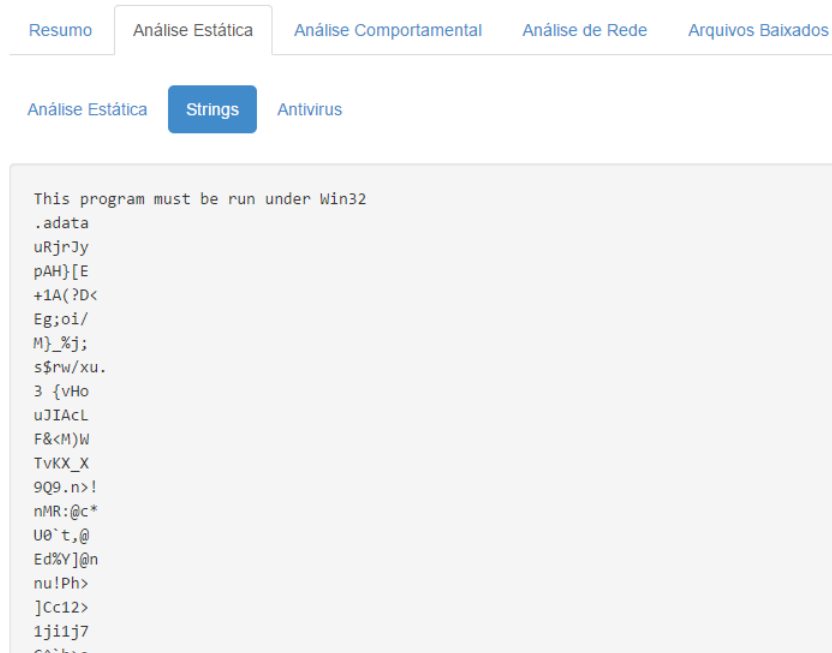


Figura 4.4: Strings extraídas do artefato

Temos ainda informações sobre bibliotecas compartilhadas do sistema e importadas pelo executável, essas bibliotecas podem indicar algumas funcionalidades do artefato analisado, por exemplo: a biblioteca “*URLDownloadToFileA*” é utilizada para baixar dados da internet e salvá-lo em um arquivo local, o que pode indicar que o artefato realizou algum download.

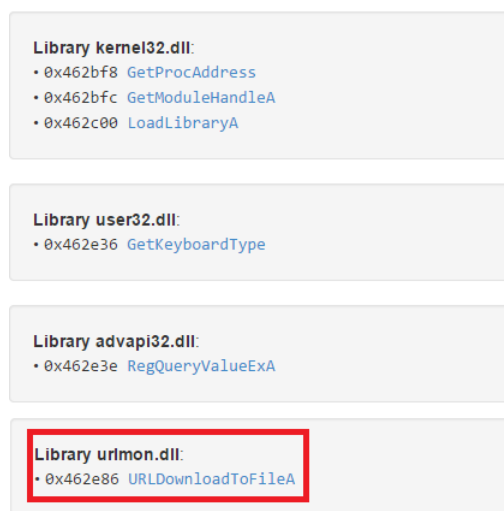


Figura 4.5: algumas funções utilizadas pelo artefato

Na figura 4.6 temos a consulta feita pelo UnBox ao VirusTotal (Serviço do google que realiza a verificação de um arquivo ou URL em 45 antivírus), verificando-se que o artefato foi identificado por vários antivírus como um cavalo de tróia (*trojan*), em alguns sendo identificado como um

trojan do tipo *downloader* (tem a função de baixar e instalar outros códigos maliciosos no sistema infectado).

Antivirus	Assinaturas
Bkav	Clean
MicroWorld-eScan	Trojan.Generic.12186100
nProtect	Trojan.Generic.12186100
CMC	Clean
CAT-QuickHeal	Clean
ALYac	Trojan.Generic.12186100
Malwarebytes	Clean
Zillya	Clean
K7AntiVirus	Riskware (0040eff71)
K7GW	Riskware (0040eff71)
TheHacker	Clean
Agnitum	Clean
Cyren	Clean
Symantec	WS.Reputation.1
Norman	Clean
TotalDefense	Clean
TrendMicro-HouseCall	Suspicious_GEN.F47V1121
Avast	Clean
ClamAV	Clean
Kaspersky	HEUR:Trojan-Downloader.Win32.Generic
BitDefender	Trojan.Generic.12186100

Figura 4.6: Consulta aos antivírus

4.1.2 Análise de rede

Na Análise de rede podemos ver os endereços acessados e detalhes dos protocolos de rede utilizados pelo artefato.

A figura 4.7 mostra os IP's e os domínios acessados. Os dois primeiros IP's (164.41.101.11 e 164.41.101.4) são servidores DNS da UnB, utilizamos para traduzir os domínios acessados pelo artefato, os IP's 189.38.90.49 e 200.98.169.40 estão atrelados aos domínios *papatudoalimentos.com.br* e *limueiro.ddns.net* respectivamente.

Hosts		Domínios	
IP		Domínios	IP
164.41.101.11		papatudoalimentos.com.br	189.38.90.49
164.41.101.4		limueiro.ddns.net	200.98.169.40
189.38.90.49			
200.98.169.40			

Figura 4.7: IPs e Domínios acessados

Foi analisado o domínio *papatudoalimentos.com.br* e na opção antivírus do UnBox podemos verificar que se trata de uma página maliciosa.

Pode-se verificar na figura 4.8 uma requisição HTTP, o que indica que ocorreu um download de um arquivo como o nome *reviera.nil* hospedado no domínio já analisado anteriormente.

Hosts (4) Domínios (2) **HTTP (1)** ICMP (0) IRC (0)

HTTP Requests

URI	Dados
http://papatudoalimentos.com.br/erros/02/reviera.nil	<pre>GET /erros/02/reviera.nil HTTP/1.1 Accept: */* Accept-Encoding: gzip, deflate User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident /4.0) Host: papatudoalimentos.com.br Connection: Keep-Alive</pre>

Figura 4.8: Protocolo HTTP

4.1.3 Arquivos Baixados

A figura 4.9 mostra detalhes de dois arquivos baixados pelo artefato, um arquivo de dados com o nome *F37694E6* e um executável com o nome *reviera[1].nil*.

Pode-se observar que o arquivo *reviera[1].nil* caiu na regra *vmdetect* do Yara o que indica que foram encontrados strings neste artefato que indicam o uso de técnicas anti-virtualização, utilizada por malwares para dificultar a análise por meio de sandboxes, como estamos tratando de um possível *downloader* é provável que esse arquivo execute as ações maliciosas.

Nome do Arquivo	reviera[1].nil
Tamanho	3392512 bytes
Tipo	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	a2f06c5e01d995be2ecfcc49a711e41
SHA1	76daf4f69541ef44254dab9058085907490bc302
SHA256	64cbbe0c15c5c885031ab63f2fd8add9d39d4f0bd3ac1f970f7b7634004e31f2
CRC32	6B6ABE0F
Ssdeep	98304:OywxJt3qAaweWGrvgtgZ9CE89fkkLRwwiOBn9FT.+PwZwErvgtks8kLRzn95
Yara	<ul style="list-style-type: none"> • vmdetect - Possibly employs anti-virtualization techniques
Download	

Nome do Arquivo	F37694E6
Tamanho	14 bytes
Tipo	data
MD5	982835d72e1b62b741be7b616389f8ba
SHA1	dbe51cb1a9fefec3655816699a6cd409724851e9
SHA256	7af8e77f5dcbedeed3df218475de49136b98601371cb3a9a385190e0946857f5f
CRC32	307DC26F
Ssdeep	3:sHQ6a/kn:sHQ67
Yara	None matched
Download	

Figura 4.9: Arquivos baixados

O UnBox permite que se faça o download dos arquivos analisados, dessa forma foi baixado o executável *reviera[1].nil* e submetido há uma nova análise. Verificamos que o artefato acessa a chave de registro "*HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\System\DNSClient*", essa chave altera o servidor DNS que o sistema irá consultar. Um atacante pode utilizar um servidor DNS para direcionar um acesso de um domínio legítimo, como por exemplo, o endereço de um banco, para uma página maliciosa. O executável foi ainda classificado como malicioso por diversos antivírus no VirusTotal. A figura 4.10 mostra algumas chaves acessadas pelo artefato e a figura 4.11 mostra a análise feita pelos antivírus.

```
HKEY_CLASSES_ROOT\CLSID\{CF4CC405-E2C5-4DDD-B3CE-5E7582D8C9FA}
HKEY_CLASSES_ROOT\CLSID\{CF4CC405-E2C5-4DDD-B3CE-5E7582D8C9FA}\TreatAs
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\System\DNSClient
HKEY_LOCAL_MACHINE\Software\Microsoft\WBEM\CIMOM
CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}
CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\TreatAs
\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}
\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\InprocServer32
\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\InprocServerX86
\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\LocalServer32
\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\InprocHandler32
```

Figura 4.10: Chaves de Registro acessadas pelo arquivo *reviera[1].nil*

Avira	TR/Crypt.XPACK.Gen
Antiy-AVL	Clean
AegisLab	Clean
AhnLab-V3	Clean
GData	Gen.Variant.Kazy.72254
ByteHero	Clean
AVware	Clean
Baidu-International	Trojan.Win32.Enigma.bAAA
Zoner	Clean
ESET-NOD32	a variant of Win32/Packed.Enigma.AAA
Rising	Clean
Ikarus	Virus.Win32.Vundo

Figura 4.11: Análise de antivírus do arquivo *reviera[1].nil*

4.1.4 Resultado do estudo de caso 1

Podemos concluir através da análise que o arquivo “Receita Federal.cpl” é um código malicioso, conforme indica a análise estática através do VírusTotal. Com a análise das bibliotecas acessadas e do detalhamento do protocolo HTTP, utilizada para o download de outro artefato pode-se classificar o código malicioso como um cavalo de troia do tipo *downloader*.

Analisando-se o artefato baixado, verificamos, através de uma chave de registro alterada, que o malware pode alterar o servidor DNS que o sistema operacional atacado utiliza, podendo assim redirecionar o tráfego de um domínio legítimo para uma página maliciosa com o objetivo de capturar informações do usuário, por exemplo. O artefato ainda apresenta padrões de strings que indicam que ele utiliza uma técnica para dificultar a análise através de analisadores sandbox.

Através desse estudo de caso podemos verificar a eficiência dos recursos apresentados pelo UnBox na identificação de um código malicioso.

4.2 Estudo de Caso 2

Baixamos o artefato *chequedevolvido.zip* disponível na URL <http://177.70.97.68/chequedevolvido.pdf> no dia 05/12/2014 extraímos e enviamos o executável *chequedevolvido.cpl*.

Na figura 4.12 temos as informações que caracterizam o arquivo como tamanho, tipo (executável do windows), *hashs* e *fuzzy hashes*.

The screenshot shows a web-based file analysis interface. At the top, there are navigation tabs: 'Resumo', 'Análise Estática', 'Análise Comportamental', 'Análise de Rede', and 'Arquivos Baixados'. The 'Análise' section is active, displaying a table with the following data:

Categoria	Início	Terminada	Duração
FILE	2014-12-06 12:47:00	2014-12-06 12:49:53	173 seconds

Below the table, the 'Especificações do artefato' section provides detailed metadata for the file 'chequedevolvido.cpl':

- Nome:** chequedevolvido.cpl
- Tamanho:** 450560 bytes
- Tipo:** PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, UPX compressed
- MD5:** 2ef057f26a589ca7a3e0bfe53cc99a06
- SHA1:** 4625a60965c2197162cbc9371244a7996005b423
- SHA256:** 67e7e9529b92d80fb2cd7a79b9f41f955b9fd06e375d27ed31628417d5876ae9
- SHA512:** 78d89bb612e37b28e8e9559767c9919623f361c7fd7b023b9c9d181fe21ebfea331b83b7c33d150baa0461cbcd3532eb8ed152daedc4063345f1c2f1b9d8cc0
- CRC32:** 60FD12C3
- Ssdeep:** 6144:sQ!l9FaRIP+dabq+yJwShca9gGhTxx3u5gkXNUIdVCM4mz7xeu6UV:sQBMgq+JS0GHMekdICM4mz7bR
- Yara:** Não encontrado

A 'Download' button is visible at the bottom of the specifications section.

Figura 4.12: Detalhes do arquivo *chequedevolvido.cpl*

O UnBox disponibiliza as capturas de tela do momento da execução, porem não há informação útil para esse artefato, pois a sua execução ocorre totalmente em segundo plano.

Podemos observar na figura 4.13 os arquivos acessados durante a execução, observa-se que o artefato acessa o arquivo de dados *uagrap.cab* que foi baixado durante a execução, conforme veremos na análise de rede.

Arquivos Chaves de registro Mutexes

```

C:\WINDOWS\system32\Shell32.dll
C:\WINDOWS\system32\Shell32.dll.123.Manifest
C:\WINDOWS\system32\msctfime.ime
C:\DOCUME~1\CLEOSV~1\CONFIG~1\Temp\chequedeveloppido.cpl
C:\DOCUME~1\CLEOSV~1\CONFIG~1\Temp\chequedeveloppido.cpl.123.Manifest
C:\SystemRoot\AppPatch\sysmain.sdb
C:\SystemRoot\AppPatch\sysrest.sdb
C:\Device\NamedPipe\ShimViewer
C:\DOCUME~1\CLEOSV~1\CONFIG~1\Temp\
chequedeveloppido.cpl
C:\DOCUME~1
C:\DOCUME~1\CLEOSV~1
C:\DOCUME~1\CLEOSV~1\CONFIG~1
C:\DOCUME~1\CLEOSV~1\CONFIG~1\Temp
C:\WINDOWS\system32\
msctfime.ime
C:\WINDOWS
C:\WINDOWS\system32
C:\
C:\WINDOWS\Registration\R000000000007.clb
C:\Documents and Settings\cleosvaldo\Dados de aplicativos\xx
PIPE\ROUTER
C:\Documents and Settings\cleosvaldo\Dados de aplicativos\tbrtnmwhnaupvhoxfociluagrap.cab

```

Figura 4.13: Arquivos acessados no Estudo de Caso 2

Na análise das chaves de registros alteradas, observa-se que o artefato acessa as chaves "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp" e "HKEY_LOCAL_MACHINE\Software\Microsoft\windows\CurrentVersion\Internet Settings\Connections" utilizadas na configuração de servidores de proxy no Windows. Com isso o artefato pode configurar um servidor de proxy malicioso no sistema com a intenção de redirecionar o acesso a do usuário para páginas falsas.

Arquivos Chaves de registro Mutexes

```

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Control Panel\Cpls
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Control Panel\Cpls
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IMM
HKEY_USERS\S-1-5-21-1078001533-688789844-725345543-1003\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers
HKEY_CURRENT_USER\SOFTWARE\Microsoft\CTF
HKEY_LOCAL_MACHINE\Software\Microsoft\CTF\SystemShared
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
HKEY_CLASSES_ROOT\CLSID\{2087C2F4-2CEF-4953-A8AB-66779B670495}\TreatAs
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp
HKEY_LOCAL_MACHINE\Software\Microsoft\windows\CurrentVersion\Internet Settings\Connections
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\WinHttp\UnsafeSslApps
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook

```

Figura 4.14: Chaves de registro

4.2.1 Análise estática

As figuras 4.15 e 4.16 mostram as bibliotecas compartilhadas acessadas e as cadeias de caracteres retiradas do artefato. Porém não foi identificada informação relevante nessa análise.

Importar

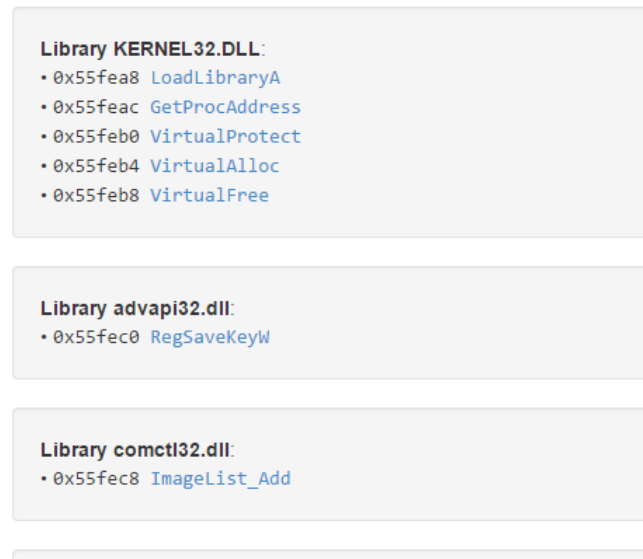


Figura 4.15: Bibliotecas acessadas na execução do artefato

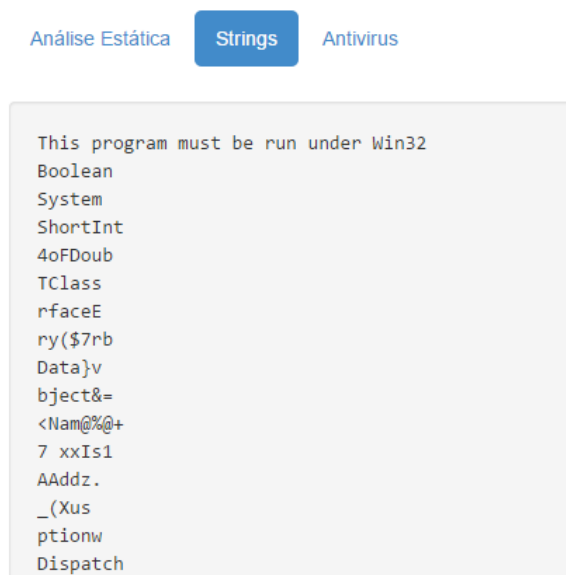


Figura 4.16: Strings retiradas do artefato

Na figura 4.17 temos a análise do VirusTotal, observa-se que ele foi classificado como um cavalo de tróia por alguns antivírus.

Microsoft	VirTool:Win32/DelfInject.gen!BI
SUPERAntiSpyware	Clean
AhnLab-V3	Clean
GData	Gen:Variant.Symmi.48691
ByteHero	Clean
ALYac	Gen:Variant.Symmi.48691
AVware	Clean
VBA32	Clean
Panda	Clean
Zoner	Clean
ESET-NOD32	a variant of Win32/TrojanDownloader.Banload.UVM
Rising	Clean
Ikarus	Trojan.Delf
Fortinet	Clean
AVG	Downloader.Banload2.UPW
Baidu-International	Trojan.Win32.Banload.BUVM
Qihoo-360	Clean

Figura 4.17: Análise de alguns antivírus

4.2.2 Análise de rede

Na análise de rede pode-se observar que o artefato se comunicou com IP 177.70.97.68 baixando um arquivo de dados com o nome *arqb.cab* e acessando uma página web, conforme mostram as figura 4.18 e 4.19.

Hosts (1)	Domínios (0)	HTTP (2)	ICMP (0)	IRC (0)
Hosts				
IP				
177.70.97.68				

Figura 4.18: IPs acessados

Hosts (1)	Domínios (0)	HTTP (2)	ICMP (0)	IRC (0)
HTTP Requests				
URI	Dados			
http://177.70.97.68//aplicativos/up/arqb.cab	<pre>GET //aplicativos/up/arqb.cab HTTP/1.1 Host: 177.70.97.68 Accept: text/html, */* Accept-Encoding: identity User-Agent: Mozilla/3.0 (compatible; Indy Library)</pre>			
http://177.70.97.68/aviso/clientes/index.php	<pre>POST /aviso/clientes/index.php HTTP/1.1 Content-Type: application/x-www-form-urlencoded Content-Length: 65 Accept: */* User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5) Host: 177.70.97.68 Connection: Keep-Alive pc=VENDAS&av=N-A&gb=N-A&wd=Windows XP (32)&dt=06:12:2014-13:27:35</pre>			

Figura 4.19: Protocolo HTTP

4.2.3 Arquivos Baixados

Na figura 4.20 vemos detalhes do arquivo *arqb.cab*, o artefato foi baixado e enviado para o UnBox para uma nova análise, porém não foi encontrado nenhuma informação relevante para a análise do artefato inicial.

Resumo	Análise Estática	Análise Comportamental	Análise de Rede	Arquivos Baixados
Nome do Arquivo	uagrap.cab			
Tamanho	2441700 bytes			
Tipo	data			
MD5	0c5154a069b55d255eb445d4d841f407			
SHA1	6beee94c7f0fe1ad0d42c81fc2718b116401154			
SHA256	605bb84884606c9d6ca1e5a7e5ed100eea38522dc5fcb6906e9970b0935241be			
CRC32	D27606EC			
Ssdeep	96:BP...PF:T			
Yara	None matched			
Download				

Figura 4.20: Arquivo baixado

4.2.4 Resultado do estudo de caso 2

Após a análise podemos concluir que o artefato *chequedevolvido.cpl* é um cavalo tróia que tem a funcionalidade de alterar chaves de registros do Windows responsáveis pela configuração

de servidores de proxy. Caso um proxy malicioso seja configurado o tráfego do usuário pode ser redirecionado para páginas falsas. Essas páginas podem capturar informações como acessos de webmail, credenciais bancárias ou até acesso a sites de fidelidade para o roubo dos pontos disponíveis nessas páginas.

Capítulo 5

Conclusão

O UnBox se mostrou uma ferramenta eficiente na identificação de códigos maliciosos, por combinar várias técnicas de análise conhecidas, o que se torna necessário devido a grande quantidade e diversidade de malwares existentes atualmente e as várias técnicas utilizadas por invasores para dificultar o reconhecimento de um código malicioso.

O UnBox foi inserido na rede da UnB com sucesso e sua integração, embora possa ser aprimorada para o contexto da Universidade, apresenta uma boa estabilidade e capacidade de usabilidade imediata, isto é, a ferramenta encontra-se operacional e acessível à toda a comunidade acadêmica e comunidade externa. Com isso, a Universidade pode contar com uma ferramenta automatizada de análise de artefatos suspeitos. É importante ressaltar que é possível realizar maiores customizações para o ambiente acadêmico, como criação de regras no Yara, criação de um banco de análises, de forma a manter análises realizadas documentadas e maior integração com os serviços oferecidos na Universidade.

Considerando as análises realizadas via UnBox dos arquivos *Receita Federal.cpl* e *chequedevolvido.cpl*, o Framework foi capaz de analisar de forma significativa os artefatos, identificando as principais características, simulando suas ações em ambiente virtualizado e reportando na forma de relatório os resultados obtidos.

É possível concluir também que a ferramenta se comporta positivamente em um ambiente real, isto é, tem o comportamento significativo e análises consideravelmente precisas dos artefatos URLs submetidos a ele. É importante ressaltar que, neste trabalho, o UnBox foi utilizado de forma extremamente genérica, ou seja, sem grandes customizações para o ambiente utilizado. a técnica *Sandbox* mostrou-se extremamente eficiente e capaz de reportar resultados realistas, de forma a apresentar-se como uma excelente alternativa para a segurança de redes de computadores. Observa-se, no entanto, que esta técnica pode não ser eficiente em casos onde o artefato não apresente sua ação de imediato, o que configura-se como *anti-sandboxing*. O UnBox mostrou-se extremamente eficiente no tocante as técnicas de análise estática, como: análise de *strings*, bibliotecas e funções importadas pelo artefato e análise por antivírus via VirusTotal. Por fim a combinação das técnicas de análise estática e da análise dinâmica via *sandbox* aumenta consideravelmente a eficiência na identificação de códigos maliciosos.

Para trabalho futuros, sugere-se:

- Criação de regras do Yara para identificação de padrões de *strings* em malwares inseridos no cenário nacional;
- Criação de mais máquinas virtuais para que ocorram mais de uma análise ao mesmo tempo no UnBox;
- Identificação de arquivos (através de *hashs*) e URLs já analisados, para que se evite um número excessivo de reanálises, poupando assim recursos do ambiente onde o UnBox esta hospedado;
- Atualizar a máquina virtual do UnBox para que as análises ocorram em uma versão mais recente do Windows;
- Ampliação da análise para outros sistemas operacionais, como: Linux, Android, IOS e OS X;
- Buscar parcerias com outros órgãos para a manutenção e ampliação do projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ABREU, R. N. P. ; CIDADE, T. F. V. (2013). Framework para análise dinâmica de códigos maliciosos. Trabalho de Conclusão de Curso em Engenharia de Redes de Comunicação, 2013, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 58p.
- [2] [BORGES; GOMES; DUARTE, 2012] César Augusto Borges de Andrade, Cláudio Gomes de Mello, Júlio Cesar Duarte. Malware Automatic Analysis (2012).
- [3] [SIKORSKI, HONIG 2012] Michael Sikorski and Andrew Honig. Practical Malware Analysis – The Hands-On Guide to Dissecting Malicious Software (2012).
- [4] [CERT.br, 2013] Estatísticas do CERT.br. Disponível em: <http://goo.gl/HK9mBT>, acessado em: 07/12/14.
- [5] [CAIS, 2013] Publicações do CAIS. Disponível em: <http://goo.gl/OpUF0n> acessado em: 07/12/14.
- [6] [PEOTTA, 2012] DAP (Dynamic Authorization Protocol): Uma Abordagem Segura Out-Of-Band Para E-Bank Com Um Segundo Fator De Autenticação Visual.
- [7] [SBSeg, 2011] Livro Minicursos do Simpósio Brasileiro de Segurança (SBSeg) – Editora Sociedade Brasileira de Computação (SBC).
- [8] [YARA MANUAL], Disponível em: <http://goo.gl/uxCw5e> acessado em: 07/12/2014.
- [9] [CUCKOO DOCUMENTATION], Disponível em: - <http://goo.gl/F2U0RP> acessado em: 11/11/2014.
- [10] [BLACKHAT USA 2013 - CUCKOO PRESENTATION] Detalhamento sobre o funcionamento do Cuckoo Sandbox. Disponível em: <http://goo.gl/JfXGh4>, acessado em : 06/12/2014.
- [11] [Malwr – Malware Analysis by Cuckoo Sandbox]. Disponível em: <https://malwr.com/>, acessado em: 29/12/2014