

Análise de Artefatos Maliciosos

Angelo Carlos M. Carvalho¹² Dario S. F. Filho¹² Luiz Otávio Duarte¹ Marcelo Carvalho Sachetin¹ Antonio Montes¹

¹CenPRA - Centro de Pesquisas Renato Archer
DSSI - Divisão de Segurança de Sistemas de Informação

²Unicamp - Universidade Estadual de Campinas
IC - Instituto de Computação

Outubro 2007

Índice

- 1 **Motivação**
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 **Fases do projeto**
 - Coleta
 - Análise
 - Classificação
 - Estudo de técnicas anti-forense
- 3 **Conclusão**
 - Principais resultados
 - Projetos em andamento

Índice

- 1 **Motivação**
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - Análise
 - Classificação
 - Estudo de técnicas anti-forense
- 3 Conclusão
 - Principais resultados
 - Projetos em andamento

O que são

Também conhecidos como malwares (malicious software), artefatos maliciosos são programas destinados a executar em um sistema de forma ilícita com o intuito de causar algum dano.

Tipos de artefatos maliciosos:

- Vírus
- Worms
- Trojan horses
- Spywares
- Bots
- Backdoors

Danos que causam

- Roubo, perda ou alteração de informações
- Paralisação de um serviço prestado
- Manipulação da máquina infectada
- Perda de arquivos ou de desempenho do computador e da rede

Danos que causam

- Artefatos maliciosos geraram perdas estimadas de cerca de US\$ 13 bilhões devido a perda de produtividade e queda de receita gerada por sistemas indisponíveis em 2006 em empresas de todo o mundo. (IDG Now! - 13 de junho de 2007)

Como se propagam

- Através de pastas compartilhadas em rede
- Arquivos em mídia removível
- Arquivo anexo em e-mail ou contido em alguma página da Internet
- Através de uma vulnerabilidade explorada de uma determinada aplicação

Crescimento no número de ataques

- Número de crackers que roubam contas bancárias cresceu 81% em relação ao ano passado devido a utilização de artefatos maliciosos (Info Online - 03 de agosto de 2007)
- Vírus para dispositivos móveis crescem mais de 1.200% em três anos (IDG Now! - 15 de junho de 2007)

Índice

- 1 **Motivação**
 - Artefatos maliciosos
 - **Objetivos do projeto**
- 2 Fases do projeto
 - Coleta
 - Análise
 - Classificação
 - Estudo de técnicas anti-forense
- 3 Conclusão
 - Principais resultados
 - Projetos em andamento

Projetos

- Desenvolver métodos de coleta automatizada de artefatos maliciosos
- Desenvolver um ambiente de análise
- Gerar relatórios e fazer a classificação dos artefatos com base no resultado da análise
- Pesquisar por tecnologias utilizadas nos artefatos capturados

Finalidade

- Identificar os objetivos dos artefatos atuais, afim de descobrir os danos que podem causar
- Analisar o comportamento dos artefatos, para descobrir formas de como evitar suas ações
- Desvendar as técnicas utilizadas, para evitar que elas funcionem

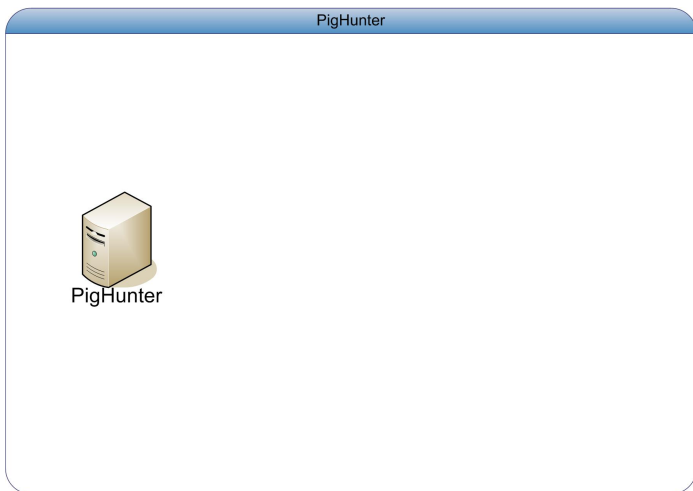
Índice

- 1 Motivação
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - Análise
 - Classificação
 - Estudo de técnicas anti-forense
- 3 Conclusão
 - Principais resultados
 - Projetos em andamento

Metodologias


- Via Blocklists
- Via SPAMS
- Através da honeynet

Via Blocklists



Via Blocklists

PigHunter

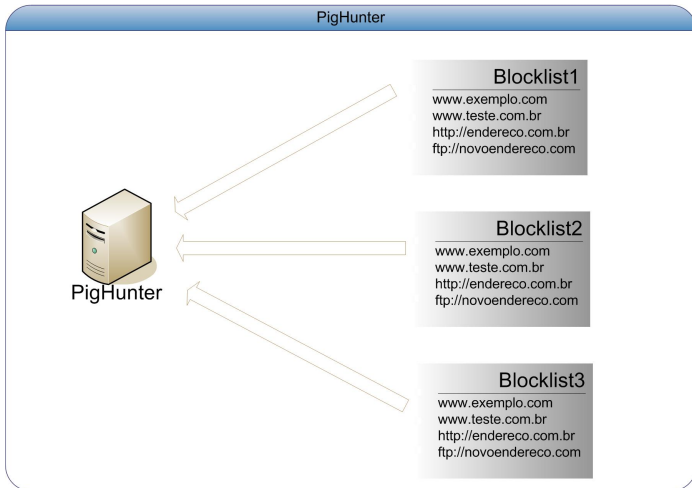


Blocklist1
www.exemplo.com
www.teste.com.br
http://endereco.com.br
ftp://novoendereco.com

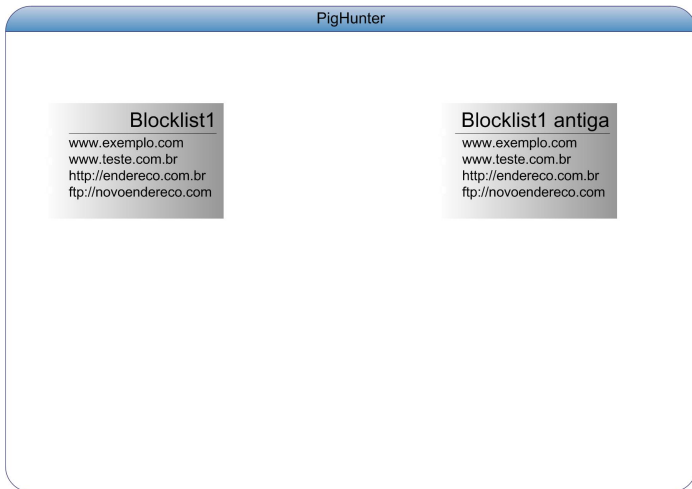
Blocklist2
www.exemplo.com
www.teste.com.br
http://endereco.com.br
ftp://novoendereco.com

Blocklist3
www.exemplo.com
www.teste.com.br
http://endereco.com.br
ftp://novoendereco.com

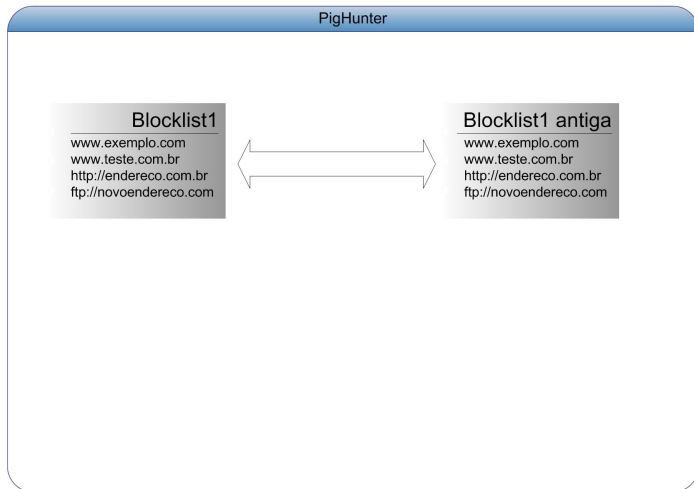
Via Blocklists



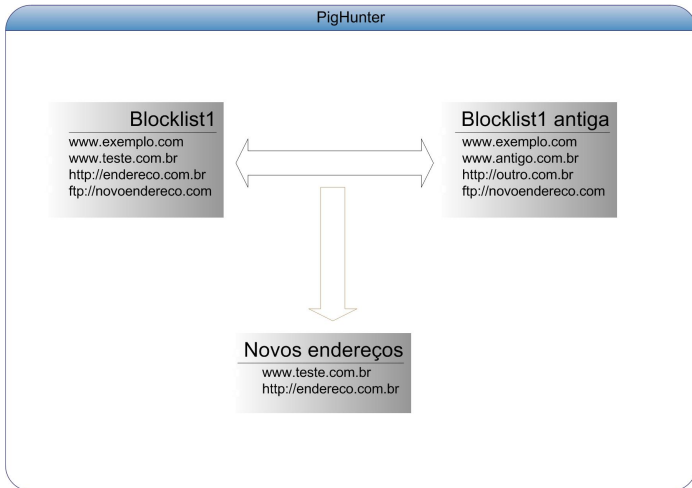
Via Blocklists



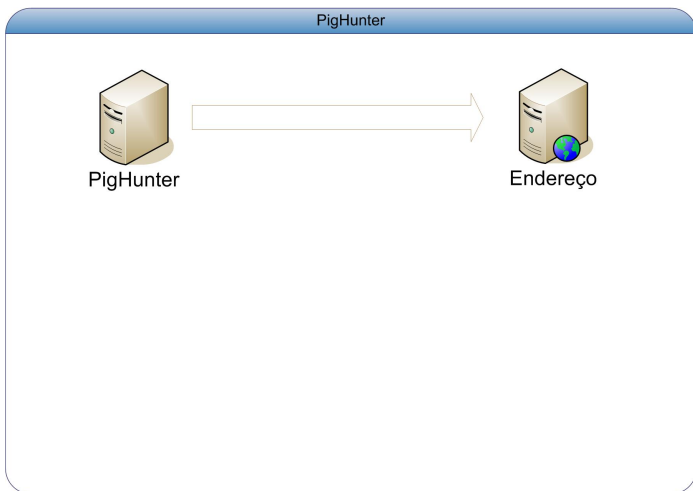
Via Blocklists



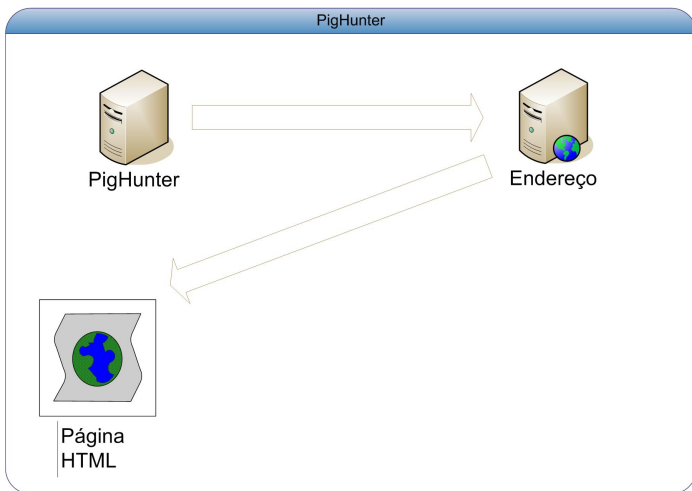
Via Blocklists



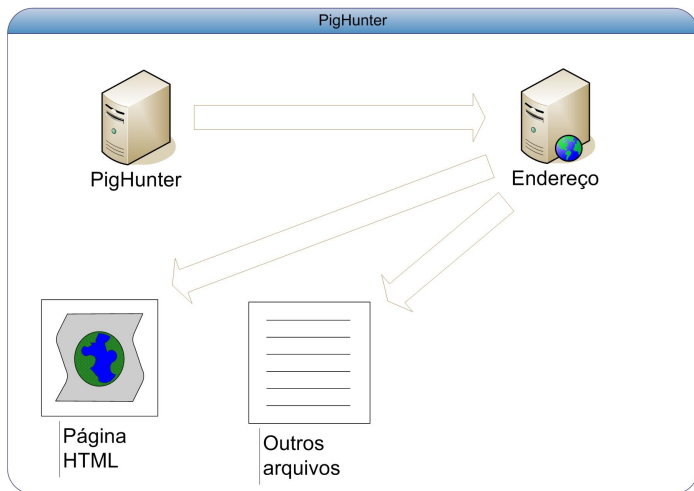
Processo de pesquisa em endereços



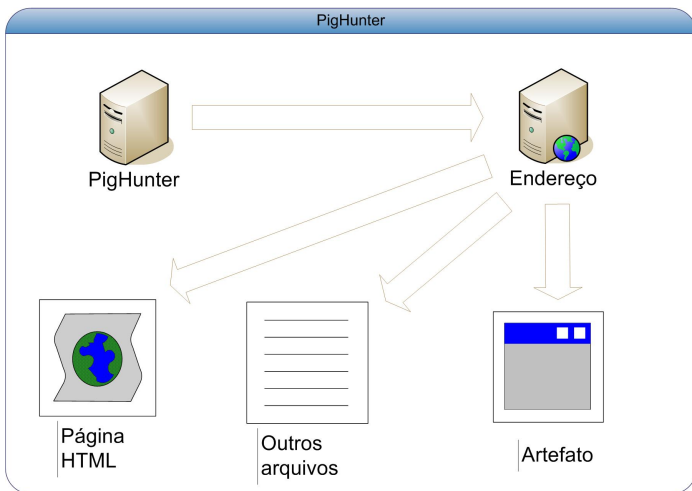
Processo de pesquisa em endereços



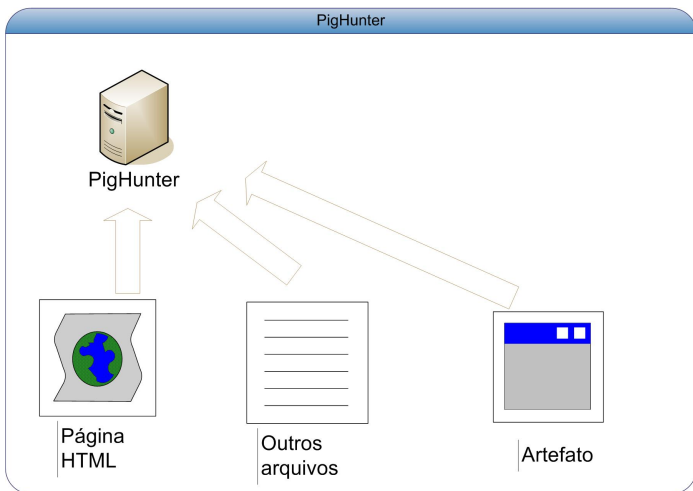
Processo de pesquisa em endereços



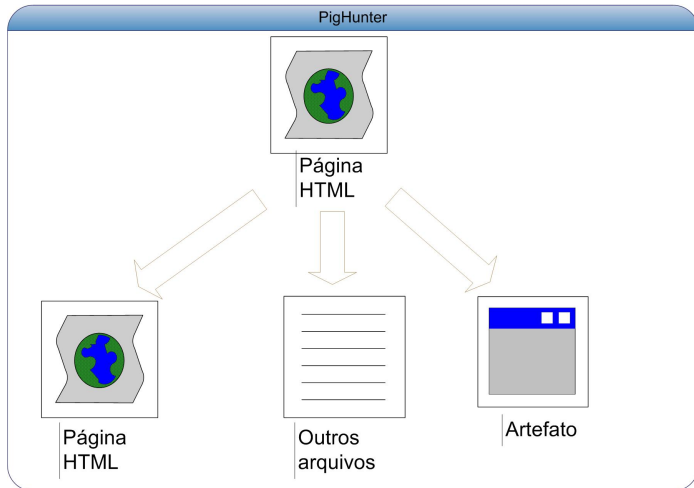
Processo de pesquisa em endereços



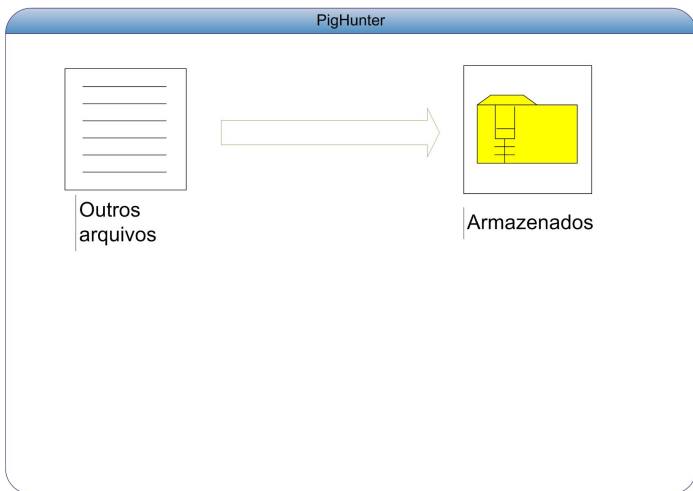
Processo de pesquisa em endereços



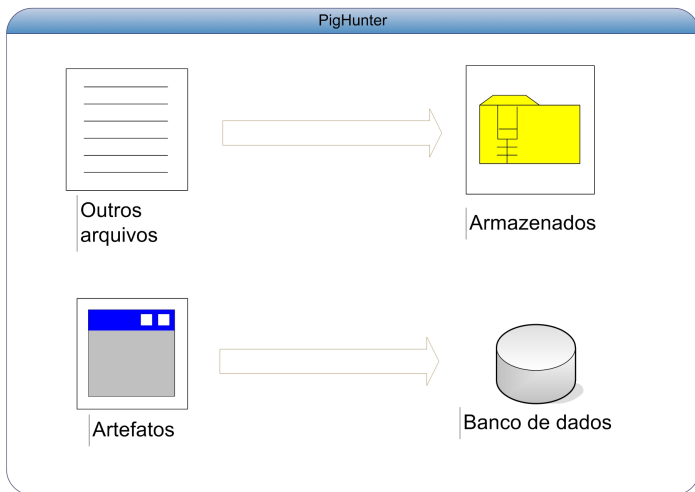
Processo de pesquisa em endereços



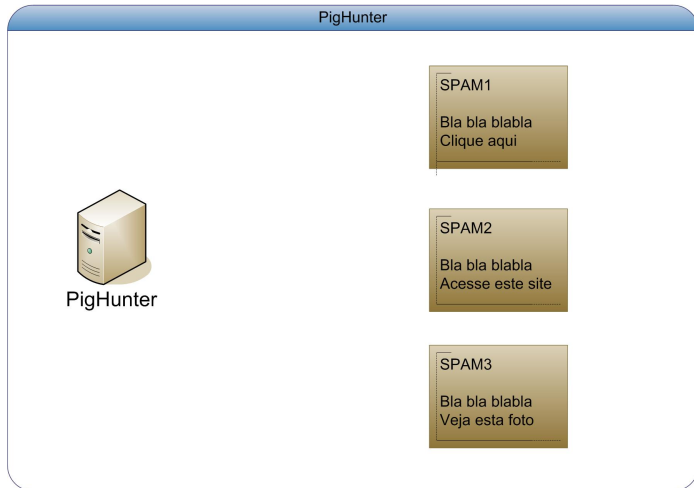
Processo de pesquisa em endereços



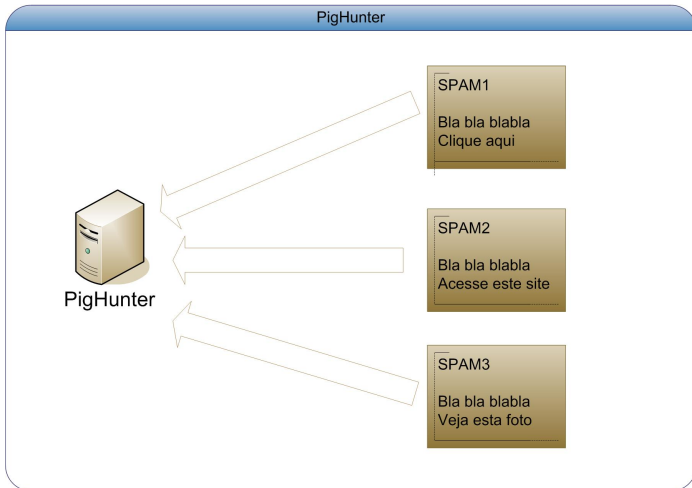
Processo de pesquisa em endereços



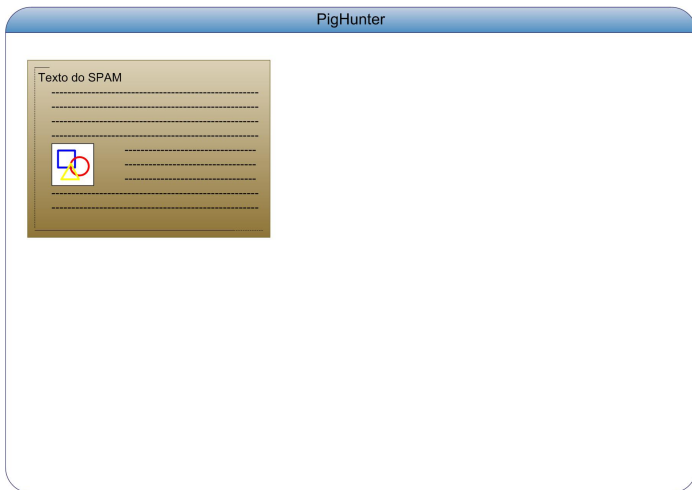
Via SPAMS



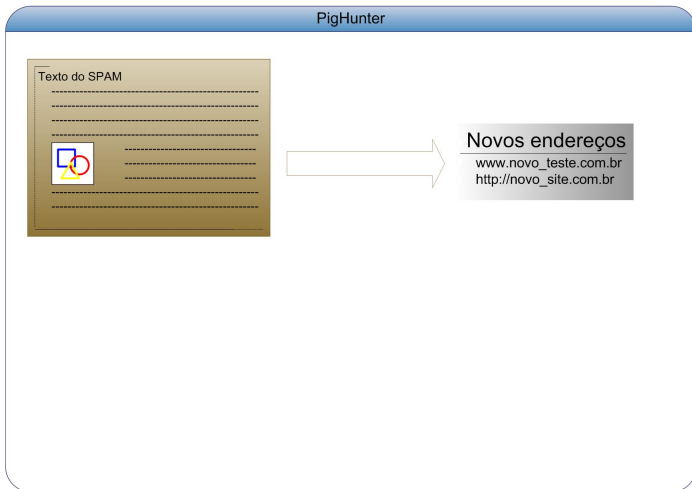
Via SPAMS



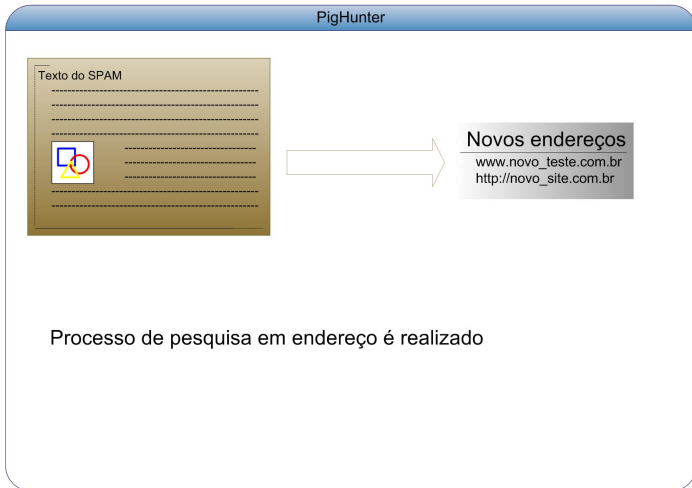
Via SPAMS



Via SPAMS



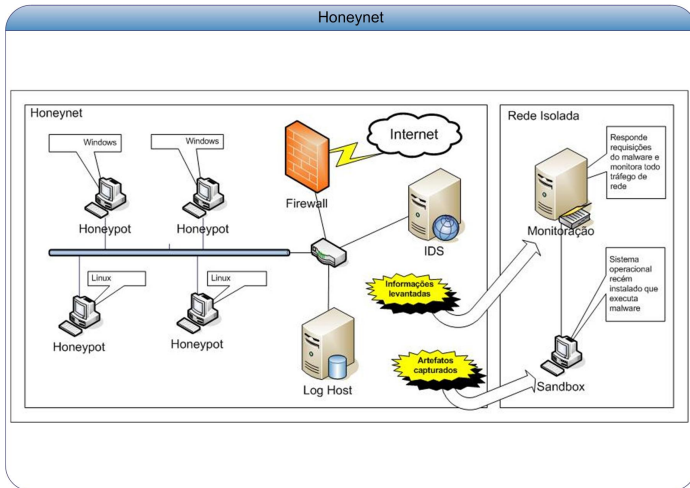
Via SPAMS



Através da honeynet

- São coletados artefatos utilizados por atacantes da Honeynet
- Os artefatos ficam armazenados nas máquinas comprometidas pelos atacantes
- O Honeypot comprometido é analisado e o resultado pode auxiliar a análise do artefato

Através da honeynet



Índice

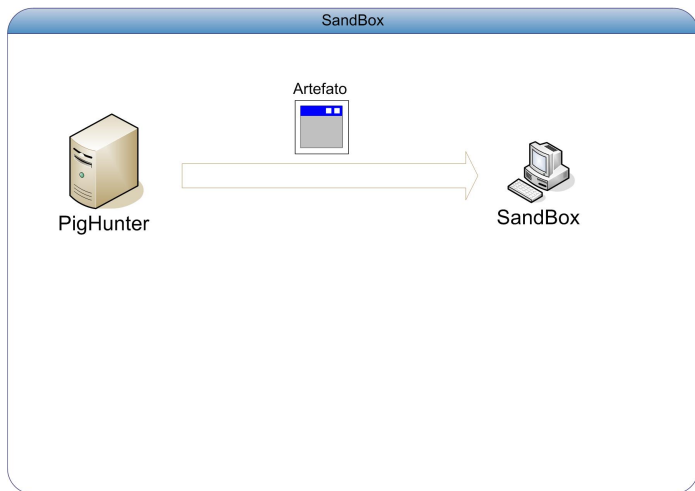
- 1 Motivação
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - **Análise**
 - Classificação
 - Estudo de técnicas anti-forense
- 3 Conclusão
 - Principais resultados
 - Projetos em andamento

Pré-análise

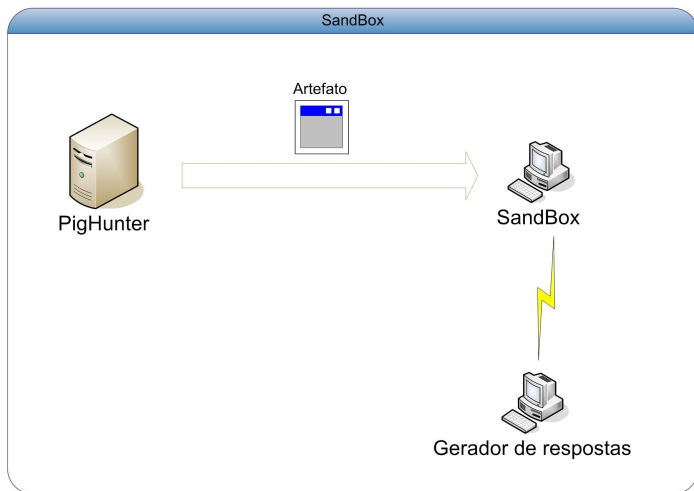
Realizada assim que o artefato é coletado
Informações coletadas:

- Tipo de artefato
- Hash
- Strings encontradas no código binário do artefato
- Packer utilizado
- DLL's que o artefato precisa para executar

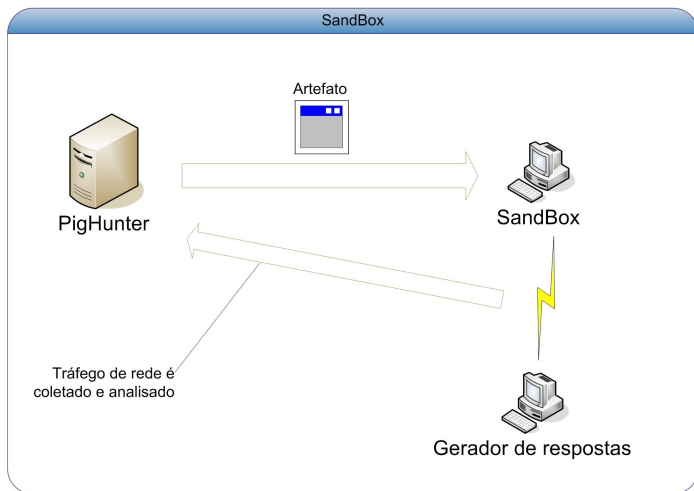
SandBox



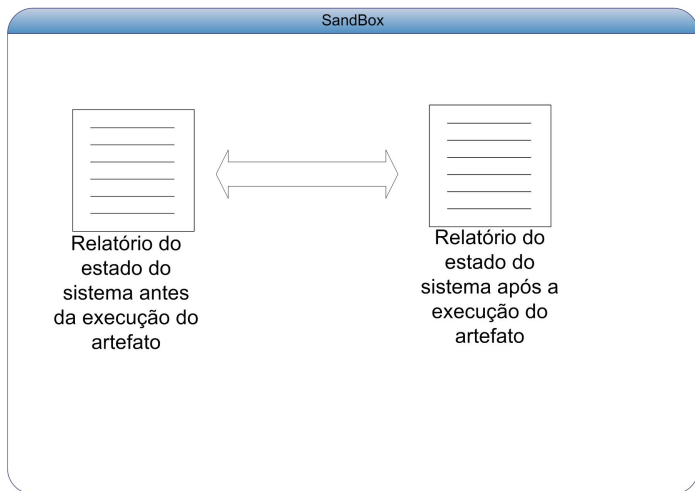
SandBox



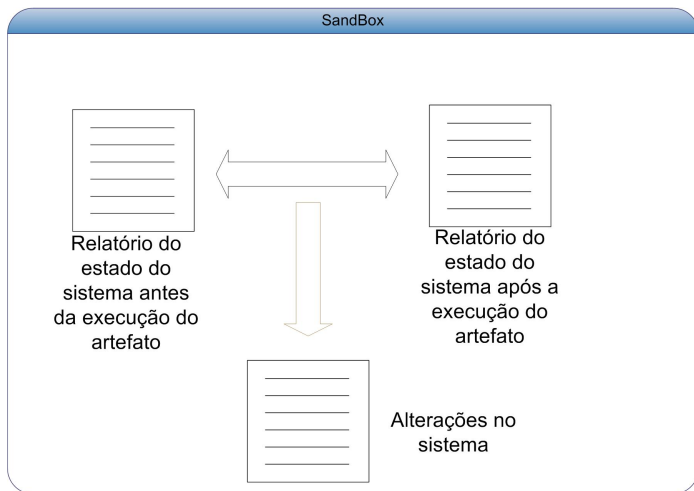
SandBox



SandBox



SandBox



Sandbox

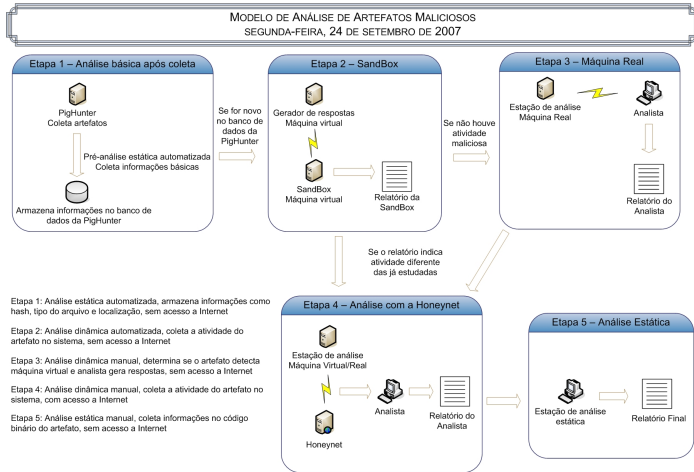
Informações coletadas:

- Tráfego de rede
- Arquivos adicionados, alterados ou excluídos do sistema
- Registro do sistema
- Programas que entraram em execução
- Portas abertas
- Programas configurados para inicializar com o sistema

Análise utilizando Honeynet

- Os artefatos que precisam de saída para a Internet, são colocados em execução na Honeynet
- Como o tráfego de saída da Honeynet é controlado, o artefato não poderá causar danos em máquinas fora da Honeynet
- Os resultados obtidos com a utilização da Honeynet na análise foram satisfatórios

Modelo de análise



Índice

- 1 Motivação
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - Análise
 - **Classificação**
 - Estudo de técnicas anti-forense
- 3 Conclusão
 - Principais resultados
 - Projetos em andamento

Tipos de artefatos maliciosos

- Trojan horse: Abre uma porta (backdoor) para manipulação remota da máquina
- Boot: Vírus que se infecta na área de inicialização dos discos rígidos
- Botclient: Se conecta a um canal de comunicação aguardando instruções

Tipos de artefatos maliciosos

- Trojan horse: Abre uma porta (backdoor) para manipulação remota da máquina
- Boot: Vírus que se infecta na área de inicialização dos discos rígidos
- Botclient: Se conecta a um canal de comunicação aguardando instruções

Tipos de artefatos maliciosos

- Trojan horse: Abre uma porta (backdoor) para manipulação remota da máquina
- Boot: Vírus que se infecta na área de inicialização dos discos rígidos
- Botclient: Se conecta a um canal de comunicação aguardando instruções

Tipos de artefatos maliciosos

- **Spyware:** Envia informações do computador infectado para outro computador
- **Keylogger:** Captura as teclas digitadas no computador
- **Worms:** Se espalham através da rede utilizando pastas compartilhadas e falhas de segurança
- **Rootkit:** Permite que o atacante tenha privilégios de administrador do computador

Tipos de artefatos maliciosos

- **Spyware:** Envia informações do computador infectado para outro computador
- **Keylogger:** Captura as teclas digitadas no computador
- **Worms:** Se espalham através da rede utilizando pastas compartilhadas e falhas de segurança
- **Rootkit:** Permite que o atacante tenha privilégios de administrador do computador

Tipos de artefatos maliciosos

- Spyware: Envia informações do computador infectado para outro computador
- Keylogger: Captura as teclas digitadas no computador
- Worms: Se espalham através da rede utilizando pastas compartilhadas e falhas de segurança
- Rootkit: Permite que o atacante tenha privilégios de administrador do computador

Tipos de artefatos maliciosos

- Spyware: Envia informações do computador infectado para outro computador
- Keylogger: Captura as teclas digitadas no computador
- Worms: Se espalham através da rede utilizando pastas compartilhadas e falhas de segurança
- Rootkit: Permite que o atacante tenha privilégios de administrador do computador

Índice

- 1 Motivação
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - Análise
 - Classificação
 - **Estudo de técnicas anti-forense**
- 3 Conclusão
 - Principais resultados
 - Projetos em andamento

Técnicas utilizadas pelos artefatos

- Detecção de máquina virtual
- Empacotamento do código binário
- Detecção do uso de depurador

Detecção de máquina virtual

- Problema: Alguns artefatos não realizavam nenhuma operação
- Motivo: Os artefatos detectavam que estavam em uma máquina virtual
- Porque: Supunham estar sobre análise e ocultavam seu funcionamento
- Solução: Alterar a parte do código que detecta a máquina virtual

Detecção de máquina virtual

- Problema: Alguns artefatos não realizavam nenhuma operação
- Motivo: Os artefatos detectavam que estavam em uma máquina virtual
- Porque: Supunham estar sobre análise e ocultavam seu funcionamento
- Solução: Alterar a parte do código que detecta a máquina virtual

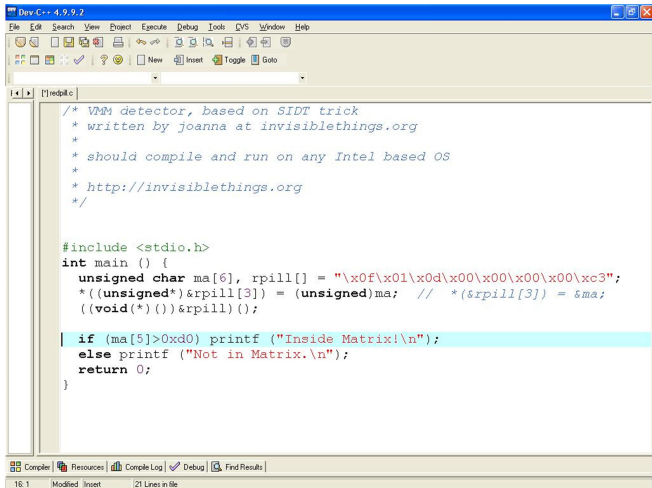
Detecção de máquina virtual

- Problema: Alguns artefatos não realizavam nenhuma operação
- Motivo: Os artefatos detectavam que estavam em uma máquina virtual
- Porque: Supunham estar sobre análise e ocultavam seu funcionamento
- Solução: Alterar a parte do código que detecta a máquina virtual

Detecção de máquina virtual

- Problema: Alguns artefatos não realizavam nenhuma operação
- Motivo: Os artefatos detectavam que estavam em uma máquina virtual
- Porque: Supunham estar sobre análise e ocultavam seu funcionamento
- Solução: Alterar a parte do código que detecta a máquina virtual

Detecção de máquina virtual



```
/* VMM detector, based on SIDT trick
 * written by joanna at invisiblethings.org
 *
 * should compile and run on any Intel based OS
 *
 * http://invisiblethings.org
 */

#include <stdio.h>
int main () {
    unsigned char ma[6], rpill[] = "\x0f\x01\x0d\x00\x00\x00\xc3";
    *((unsigned*)&rpill[3]) = (unsigned)ma; // *(&rpill[3]) = &ma;
    ((void(*)())&rpill)();

    if (ma[5]>0xd0) printf ("Inside Matrix!\n");
    else printf ("Not in Matrix.\n");
    return 0;
}
```

Detecção de máquina virtual

The screenshot shows the IDA Pro interface with the following assembly code and control flow graph:

```
mov     eax, ds:duord_403004
mov     [ebp+var_24], eax
movzx   eax, ds:byte_403008
mov     [ebp+var_20], al
lea     eax, [ebp+var_18]
mov     [ebp+var_28+3], eax
lea     eax, [ebp+var_28]
call    eax
cnp     [ebp+var_13], 000h
jbe     short loc_4012F3

loc_4012F3:
mov     [esp+40h+var_48], offset aInsideMatrix ; "Inside Matrix!\n"
call    printf
jmp     short loc_4012FF

loc_4012FF:
mov     eax, 0
leave
retn
_main endp
```

The control flow graph shows a conditional jump from the assembly code to `loc_4012F3` if the condition is not below or equal (`jbe`). Both paths lead to `loc_4012FF`, which then returns from the `_main` function.

Detecção de máquina virtual

OllyDbg - redpill.exe - [CPU - main thread, module redpill]

Registers (FPU)

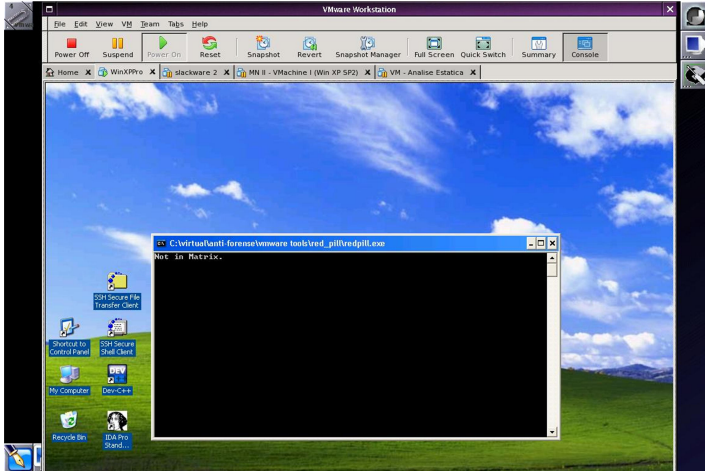
EAX	00000000	msvcrt.77C4188F
ECX	77C4188F	msvcrt.77C41870
EDX	77C41870	msvcrt.77C41870
ESI	00000000	
ESP	0025F770	
EBP	0025F770	
EIP	FFFFFFF7	
EFL	7C910738	ntdll.7C910738
EIP	0040125F	redpill.0040125F

Disassembly

Address	Hex dump	ASCII
0040125B	CALL EBX, EDI	
0040125C	CALL EBX, EDI	
0040125D	CALL EBX, EDI	
0040125E	CALL EBX, EDI	
0040125F	CALL EBX, EDI	
00401260	CALL EBX, EDI	
00401261	CALL EBX, EDI	
00401262	CALL EBX, EDI	
00401263	CALL EBX, EDI	
00401264	CALL EBX, EDI	
00401265	CALL EBX, EDI	
00401266	CALL EBX, EDI	
00401267	CALL EBX, EDI	
00401268	CALL EBX, EDI	
00401269	CALL EBX, EDI	
0040126A	CALL EBX, EDI	
0040126B	CALL EBX, EDI	
0040126C	CALL EBX, EDI	
0040126D	CALL EBX, EDI	
0040126E	CALL EBX, EDI	
0040126F	CALL EBX, EDI	
00401270	CALL EBX, EDI	
00401271	CALL EBX, EDI	
00401272	CALL EBX, EDI	
00401273	CALL EBX, EDI	
00401274	CALL EBX, EDI	
00401275	CALL EBX, EDI	
00401276	CALL EBX, EDI	
00401277	CALL EBX, EDI	
00401278	CALL EBX, EDI	
00401279	CALL EBX, EDI	
0040127A	CALL EBX, EDI	
0040127B	CALL EBX, EDI	
0040127C	CALL EBX, EDI	
0040127D	CALL EBX, EDI	
0040127E	CALL EBX, EDI	
0040127F	CALL EBX, EDI	
00401280	CALL EBX, EDI	
00401281	CALL EBX, EDI	
00401282	CALL EBX, EDI	
00401283	CALL EBX, EDI	
00401284	CALL EBX, EDI	
00401285	CALL EBX, EDI	
00401286	CALL EBX, EDI	
00401287	CALL EBX, EDI	
00401288	CALL EBX, EDI	
00401289	CALL EBX, EDI	
0040128A	CALL EBX, EDI	
0040128B	CALL EBX, EDI	
0040128C	CALL EBX, EDI	
0040128D	CALL EBX, EDI	
0040128E	CALL EBX, EDI	
0040128F	CALL EBX, EDI	
00401290	CALL EBX, EDI	
00401291	CALL EBX, EDI	
00401292	CALL EBX, EDI	
00401293	CALL EBX, EDI	
00401294	CALL EBX, EDI	
00401295	CALL EBX, EDI	
00401296	CALL EBX, EDI	
00401297	CALL EBX, EDI	
00401298	CALL EBX, EDI	
00401299	CALL EBX, EDI	
0040129A	CALL EBX, EDI	
0040129B	CALL EBX, EDI	
0040129C	CALL EBX, EDI	
0040129D	CALL EBX, EDI	
0040129E	CALL EBX, EDI	
0040129F	CALL EBX, EDI	
004012A0	CALL EBX, EDI	
004012A1	CALL EBX, EDI	
004012A2	CALL EBX, EDI	
004012A3	CALL EBX, EDI	
004012A4	CALL EBX, EDI	
004012A5	CALL EBX, EDI	
004012A6	CALL EBX, EDI	
004012A7	CALL EBX, EDI	
004012A8	CALL EBX, EDI	
004012A9	CALL EBX, EDI	
004012AA	CALL EBX, EDI	
004012AB	CALL EBX, EDI	
004012AC	CALL EBX, EDI	
004012AD	CALL EBX, EDI	
004012AE	CALL EBX, EDI	
004012AF	CALL EBX, EDI	
004012B0	CALL EBX, EDI	
004012B1	CALL EBX, EDI	
004012B2	CALL EBX, EDI	
004012B3	CALL EBX, EDI	
004012B4	CALL EBX, EDI	
004012B5	CALL EBX, EDI	
004012B6	CALL EBX, EDI	
004012B7	CALL EBX, EDI	
004012B8	CALL EBX, EDI	
004012B9	CALL EBX, EDI	
004012BA	CALL EBX, EDI	
004012BB	CALL EBX, EDI	
004012BC	CALL EBX, EDI	
004012BD	CALL EBX, EDI	
004012BE	CALL EBX, EDI	
004012BF	CALL EBX, EDI	
004012C0	CALL EBX, EDI	
004012C1	CALL EBX, EDI	
004012C2	CALL EBX, EDI	
004012C3	CALL EBX, EDI	
004012C4	CALL EBX, EDI	
004012C5	CALL EBX, EDI	
004012C6	CALL EBX, EDI	
004012C7	CALL EBX, EDI	
004012C8	CALL EBX, EDI	
004012C9	CALL EBX, EDI	
004012CA	CALL EBX, EDI	
004012CB	CALL EBX, EDI	
004012CC	CALL EBX, EDI	
004012CD	CALL EBX, EDI	
004012CE	CALL EBX, EDI	
004012CF	CALL EBX, EDI	
004012D0	CALL EBX, EDI	
004012D1	CALL EBX, EDI	
004012D2	CALL EBX, EDI	
004012D3	CALL EBX, EDI	
004012D4	CALL EBX, EDI	
004012D5	CALL EBX, EDI	
004012D6	CALL EBX, EDI	
004012D7	CALL EBX, EDI	
004012D8	CALL EBX, EDI	
004012D9	CALL EBX, EDI	
004012DA	CALL EBX, EDI	
004012DB	CALL EBX, EDI	
004012DC	CALL EBX, EDI	
004012DD	CALL EBX, EDI	
004012DE	CALL EBX, EDI	
004012DF	CALL EBX, EDI	
004012E0	CALL EBX, EDI	
004012E1	CALL EBX, EDI	
004012E2	CALL EBX, EDI	
004012E3	CALL EBX, EDI	
004012E4	CALL EBX, EDI	
004012E5	CALL EBX, EDI	
004012E6	CALL EBX, EDI	
004012E7	CALL EBX, EDI	
004012E8	CALL EBX, EDI	
004012E9	CALL EBX, EDI	
004012EA	CALL EBX, EDI	
004012EB	CALL EBX, EDI	
004012EC	CALL EBX, EDI	
004012ED	CALL EBX, EDI	
004012EE	CALL EBX, EDI	
004012EF	CALL EBX, EDI	
004012F0	CALL EBX, EDI	
004012F1	CALL EBX, EDI	
004012F2	CALL EBX, EDI	
004012F3	CALL EBX, EDI	
004012F4	CALL EBX, EDI	
004012F5	CALL EBX, EDI	
004012F6	CALL EBX, EDI	
004012F7	CALL EBX, EDI	
004012F8	CALL EBX, EDI	
004012F9	CALL EBX, EDI	
004012FA	CALL EBX, EDI	
004012FB	CALL EBX, EDI	
004012FC	CALL EBX, EDI	
004012FD	CALL EBX, EDI	
004012FE	CALL EBX, EDI	
004012FF	CALL EBX, EDI	

Breakpoint at redpill.0040125F

Detecção de máquina virtual



Empacotamento do código binário

Funções:

- Dificultar a compreensão do código assembly
- Diminuir o tamanho do artefato

Empacotamento do código binário

Exemplos de packers:

- UPX
- ASPack
- PECompact
- yoda's Crypter
- tELock
- Alguns compiladores

Empacotamento do código binário

The screenshot displays the IDA Pro interface with a control flow graph (CFG) for the file 'C:\virtual\anti-forense\vmware\tools\vmtoolsd\vmtoolsd.exe'. The graph consists of numerous nodes, each representing a basic block of code, connected by edges indicating control flow. The nodes are color-coded and labeled with addresses and assembly instructions. A 'Graph overview' window is open in the bottom right corner, providing a smaller-scale view of the entire graph. The status bar at the bottom shows the current address as 32.76% (624,10209) (644,1) 00002FFF 00412FFF: 0b_412A70+50F. The console window at the bottom displays the following text:

```
File 'C:\virtual\anti-forense\vmware\tools\vmtoolsd\vmtoolsd.exe' is successfully loaded into the database.  
Compiling file 'C:\Program Files\IDA\idc\ida.idc'...  
Executing function 'main'...  
Compiling file 'C:\Program Files\IDA\idc\onload.idc'...  
Executing function 'onload'...  
IDA is analyzing the input file...  
You may start to explore the input file right now.  
Propagating type information...  
Function argument information is propagated.  
The initial autoanalysis has been finished.
```

Detecção do uso de depurador

Funções de um depurador:

- Execução passo a passo de um programa
- Valores de variáveis em um determinado instante da execução
- Laços de código que o programa executa e quantas vezes ele executa
- Resultados dos testes condicionais

Detecção do uso de depurador

Como a detecção é feita:

- Utilizando a função `isDebuggerPresent()`
- Utilizando um processo protetor

Detecção do uso de depurador

Programa que utiliza a função `IsDebuggerPresent()`:

```
#include ( windows.h )  
int main ()  
{  
if (IsDebuggerPresent())  
printf("Está sendo executado em um depurador");  
else  
printf("Não está sendo executado em um depurador");  
return 0;  
}
```

Detecção do uso de depurador

Solução:

- Editar o código assembly alterando a resposta do teste condicional

Detecção do uso de depurador

Programa que utiliza um processo protetor:

- Para depurar um programa é necessário utilizar uma porta específica
- Um programa em execução pode ser depurado por apenas um depurador

Detecção do uso de depurador

Programa que utiliza um processo protetor:

- Para depurar um programa é necessário utilizar uma porta específica
- Um programa em execução pode ser depurado por apenas um depurador

Detecção do uso de depurador

O que o artefato faz:

- Cria um processo para depurar ele mesmo
- Outro processo que tentar depurá-lo não irá conseguir

Detecção do uso de depurador

O que o artefato faz:

- Cria um processo para depurar ele mesmo
- Outro processo que tentar depurá-lo não irá conseguir

Detecção do uso de depurador

O que o analista faz:

- Altera o código assembly

Índice

- 1 Motivação
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - Análise
 - Classificação
 - Estudo de técnicas anti-forense
- 3 **Conclusão**
 - **Principais resultados**
 - Projetos em andamento

Principais resultados.

- Coleta de mais de 3.300 artefatos
- Análise automatizada e geração de relatórios (uma análise a cada 7 minutos)
- Desenvolvimento de técnicas para analisar artefatos que utilizam novas tecnologias

Principais resultados.

- Coleta de mais de 3.300 artefatos
- Análise automatizada e geração de relatórios (uma análise a cada 7 minutos)
- Desenvolvimento de técnicas para analisar artefatos que utilizam novas tecnologias

Principais resultados.

- Coleta de mais de 3.300 artefatos
- Análise automatizada e geração de relatórios (uma análise a cada 7 minutos)
- Desenvolvimento de técnicas para analisar artefatos que utilizam novas tecnologias

Índice

- 1 Motivação
 - Artefatos maliciosos
 - Objetivos do projeto
- 2 Fases do projeto
 - Coleta
 - Análise
 - Classificação
 - Estudo de técnicas anti-forense
- 3 Conclusão
 - Principais resultados
 - **Projetos em andamento**

Projetos em andamento

- Aperfeiçoamento dos scripts desenvolvidos e das ferramentas criadas
- Desenvolvimento de ambientes de análise para tipos de artefatos específicos
- Pesquisa por novas ferramentas, tecnologias e métodos de coleta

Apoio:



UNICAMP



CenPRA
Centro de Pesquisas
Renato Archer

Contato: angelocmcarvalho@gmail.com
Dario.Fernandes@cenpra.gov.br