# Homemade Sandbox

## Como construir um analisador de malware para responder rapidamente à sua vítima

Victor Furuse Martins [1,2]
Dario Fernandes [1,2] e André Grégio [1,2]

[1]CTI - Centro de Tecnologia da Informação
[2]UNICAMP - Universidade Estadual de Campinas

5 de dezembro de 2009

Centro de Tecnologia da Informação Renato Archer

# Cronograma

# Cenário do Brasil

## Relatório de Inteligência de Segurança da Microsoft - 11/2008

*"... no primeiro semestre de 2008, o Brasil foi o sexto país mais atacado por programas maliciosos..."*

## Relatório de Ameaças à Segurança na Internet - 04/2009

*"... 90% das ameaças visavam roubar dados confidenciais..."*
*"... Brasil possui 34% das atividades maliciosas na América Latina..."*

# Como se proteger de programas maliciosos?

Proteções mais usuais:

- Anti-vírus;
- Firewall;
- Atualização constante dos programas.

**Estudo e análise do malware**

Uso de Sandbox!

# Como se proteger de programas maliciosos?

Proteções mais usuais:

- Anti-vírus;
- Firewall;
- Atualização constante dos programas.

## Estudo e análise do malware

Uso de Sandbox!

# O que é uma Sandbox?

Características:

- Ambiente controlado;
- Fácil restauração;
- Muito utilizado na análise de artefatos maliciosos.

# Sandbox na análise de artefatos maliciosos

Através das informações geradas é possível:

- Preencher lacunas da análise estática;
- Entender o funcionamento de um malware;
  - Mecanismos usados;
- Como o malware compromete a máquina;
- Saber a finalidade do malware.

# Exemplo de Sandbox existentes

Desenvolvidas e mantidas por:

## Universidades

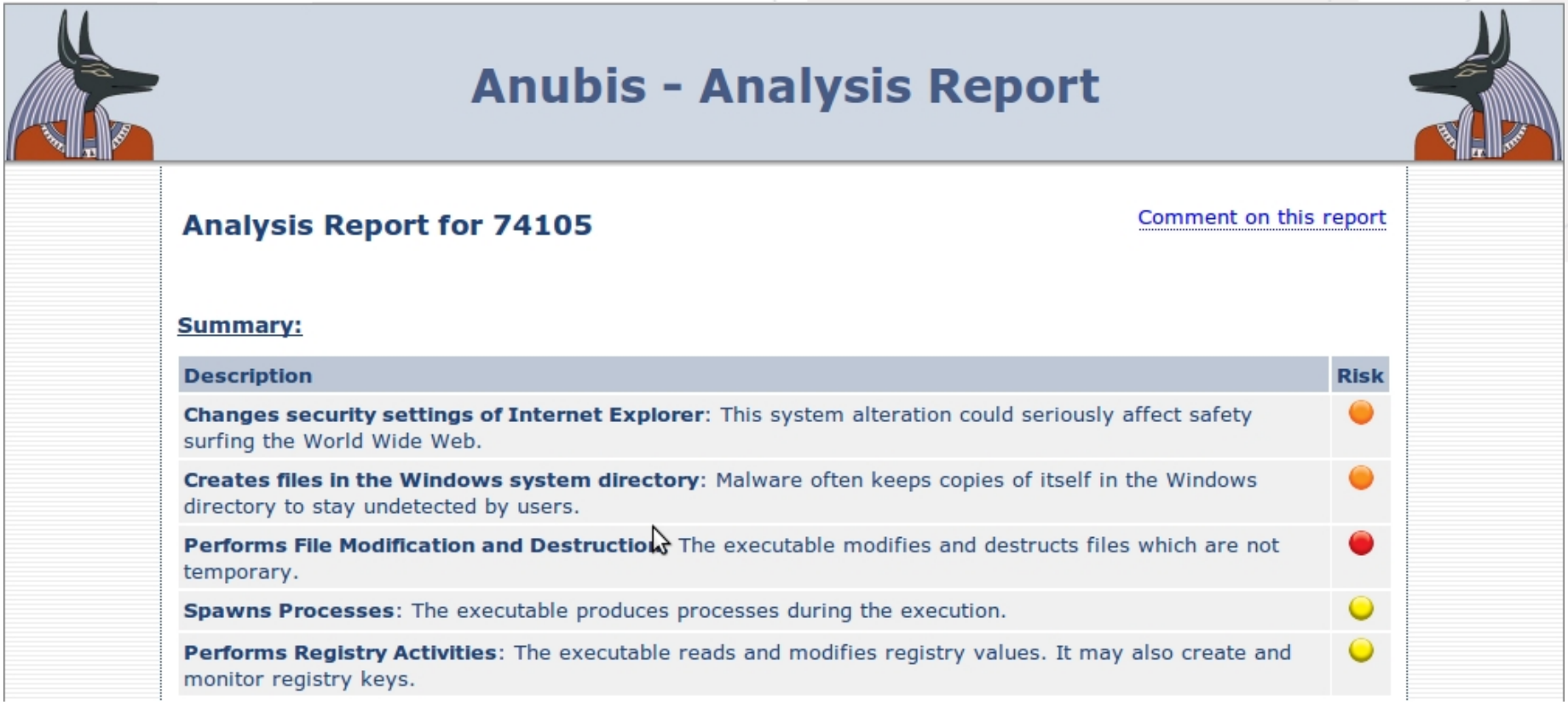
- Anubis
- CWSandbox

## Empresas privadas

- ThreatExpert
- Joebox

# Sandbox Existentes - Anubis

Exemplo de análise:



## Anubis - Analysis Report

### Analysis Report for 74105

Comment on this report

**Summary:**

| Description | Risk |
|---|---|
| **Changes security settings of Internet Explorer**: This system alteration could seriously affect safety surfing the World Wide Web. | |
| **Creates files in the Windows system directory**: Malware often keeps copies of itself in the Windows directory to stay undetected by users. | |
| **Performs File Modification and Destruction**: The executable modifies and destructs files which are not temporary. | |
| **Spawns Processes**: The executable produces processes during the execution. | |
| **Performs Registry Activities**: The executable reads and modifies registry values. It may also create and monitor registry keys. | |

# Por que devemos montar uma sandbox?

Motivações:

- Todas são serviços online;
- Relatórios fornecidos são genéricos;
  - Somente mediante contratos de serviço $\Rightarrow$ análise personalizada;
- Fila de espera;
- Limite no tamanho do arquivo enviado.

**Homemade Sandbox**

Suprir essas necessidades!

# Por que devemos montar uma sandbox?

Motivações:

- Todas são serviços online;
- Relatórios fornecidos são genéricos;
  - Somente mediante contratos de serviço $\Rightarrow$ análise personalizada;
- Fila de espera;
- Limite no tamanho do arquivo enviado.

## Homemade Sandbox

Suprir essas necessidades!

# Cronograma



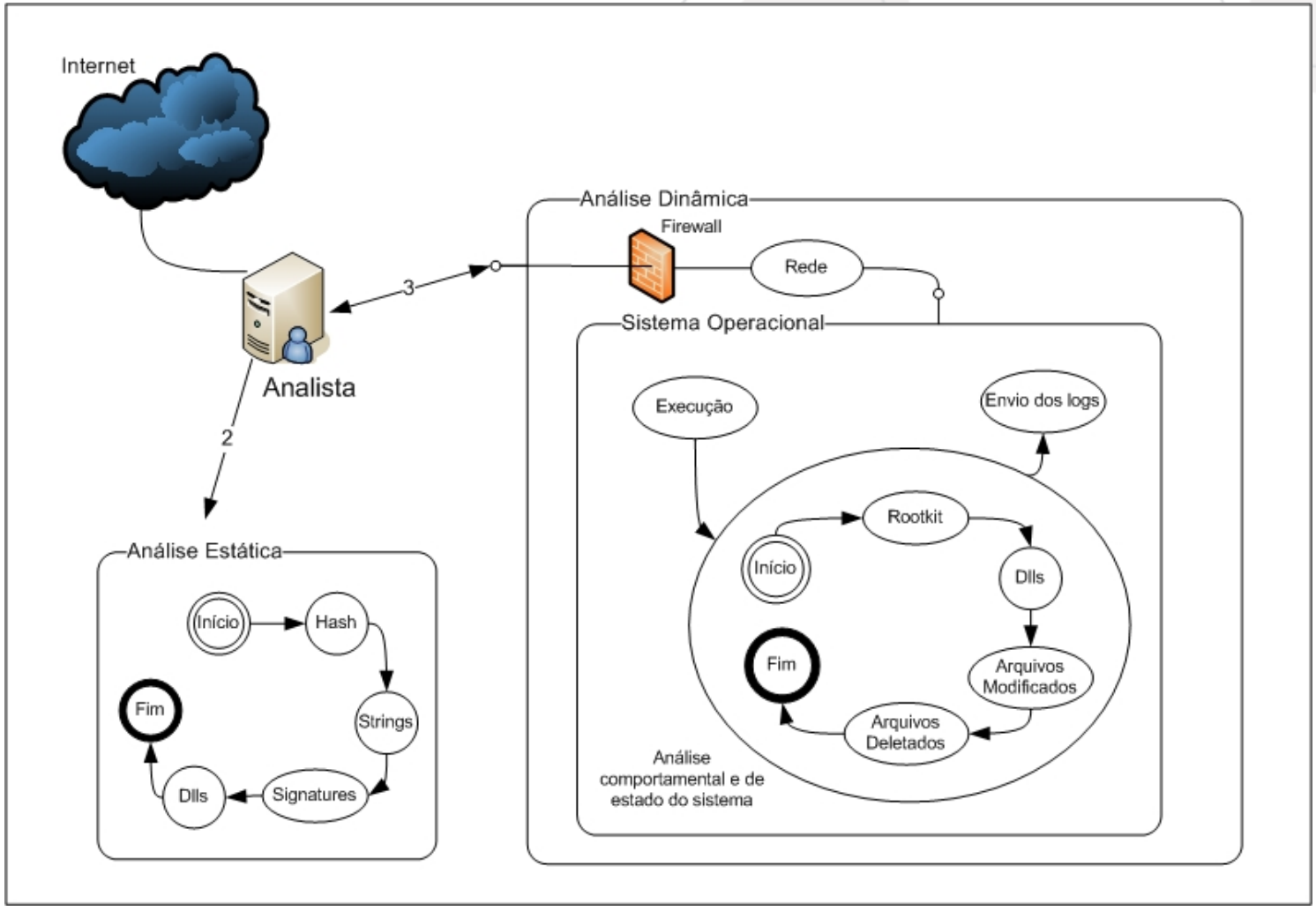
Centro de Tecnologia da Informação Renato Archer

# Atenção!

## Algumas ressalvas sobre o ambiente de análise proposto

- NÃO é indicado para ser posto em produção;
- NÃO tem proteção contra detecção de ambientes virtuais;
- Deve-se ter certeza que o ambiente está isolado.

# Diagrama do modelo proposto



# Máquina do Analista

Dividido em duas partes:

## Máquina base é responsável por:

- Efetuar a análise estática;
- Enviar o malware para a sandbox;
- Isolar o ambiente;

## Máquina virtual é responsável por:

- Efetuar a análise dinâmica;
- Ajudar a isolar o ambiente;

**Especificações da máquina** $\Rightarrow$ desktop doméstico!

# Máquina do Analista

Dividido em duas partes:

## Máquina base é responsável por:

- Efetuar a análise estática;
- Enviar o malware para a sandbox;
- Isolar o ambiente;

## Máquina virtual é responsável por:

- Efetuar a análise dinâmica;
- Ajudar a isolar o ambiente;

**Especificações da máquina** ⇒ desktop doméstico!

# Máquina do Analista

Dividido em duas partes:

## Máquina base é responsável por:

- Efetuar a análise estática;
- Enviar o malware para a sandbox;
- Isolar o ambiente;

## Máquina virtual é responsável por:

- Efetuar a análise dinâmica;
- Ajudar a isolar o ambiente;

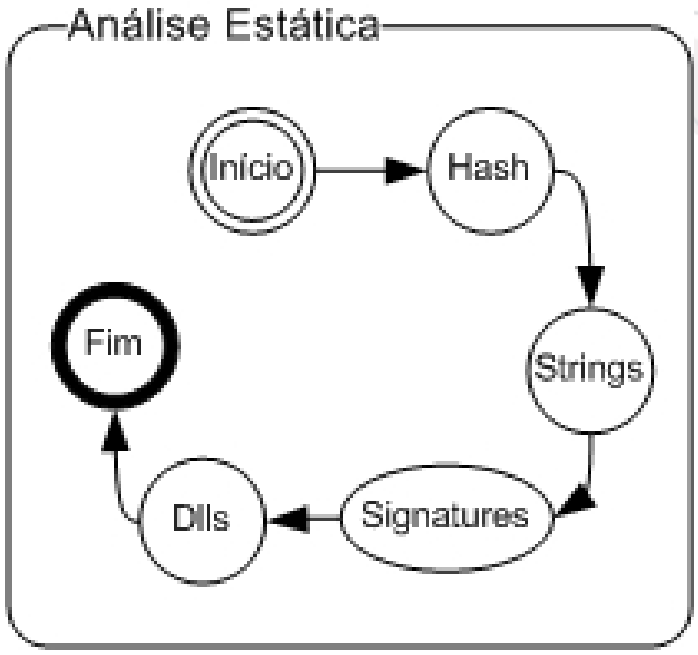
**Especificações da máquina** $\Rightarrow$ desktop doméstico!

# Cronograma



Centro de Tecnologia da Informação Renato Archer

# Ciclo da análise estática




## Tempo gasto

Cerca de 10 minutos.

# Hash e Strings

## Hash ⇒ Identificar unicamente cada malware

**md5deep** - md5sum, sha256sum ou sha512sum

## Strings ⇒ Busca por sequências legíveis

**Strings v2.41** com no mínimo 3 bytes.

# Hash e Strings

## Hash ⇒ Identificar unicamente cada malware

**md5deep** - md5sum, sha256sum ou sha512sum

## Strings ⇒ Busca por sequências legíveis

**Strings v2.41** com no mínimo 3 bytes.

# Strings

## O que procurar no log das strings?

- Senhas e usuários;
- Endereços IPs;
- Nome de funções;
- Expressões regulares;

## Exemplo de strings de malware com Armadillo v1.71

*RegSetValueExA, RegCloseKey, RegOpenKeyExA, CryptGetHashParam, CryptDestroyHash, CryptReleaseContext, CryptHashData, CryptCreateHash, CryptAcquireContextA, AdjustTokenPrivileges, LookupPrivilegeValueA, OpenProcessToken, RegCreateKeyA,*

*RegQueryValueExA, GetStartupInfoA, system32, SeDebugPrivilege, ...*

# Strings

## O que procurar no log das strings?

- Senhas e usuários;
- Endereços IPs;
- Nome de funções;
- Expressões regulares;

## Exemplo de strings de malware com Armadillo v1.71

*RegSetValueExA, RegCloseKey, RegOpenKeyExA, CryptGetHashParam, CryptDestroyHash, CryptReleaseContext, CryptHashData, CryptCreateHash, CryptAcquireContextA, AdjustTokenPrivileges, LookupPrivilegeValueA, OpenProcessToken, RegCreateKeyA,*

*RegQueryValueExA, GetStartupInfoA, system32, SeDebugPrivilege, ...*

# Signatures e DIls

## Signatures ⇒ Identificar assinaturas e packers

**Sigcheck** - verifica a presença de assinaturas da Microsoft.

- Verificar se as bibliotecas/programas nativos são legítimos.

**pefile** - módulo de Python que lê PE headers.

- Tentar remover o packer do malware.

## Load-time DIls ⇒ Identifica as bibliotecas carregadas no início do programa

**Dependency Walker**

- Presença de bibliotecas suspeitas ou falsas, usar Sigcheck.

# Signatures e DIls

## Signatures ⇒ Identificar assinaturas e packers

**Sigcheck** - verifica a presença de assinaturas da Microsoft.

- Verificar se as bibliotecas/programas nativos são legítimos.

**pefile** - módulo de Python que lê PE headers.

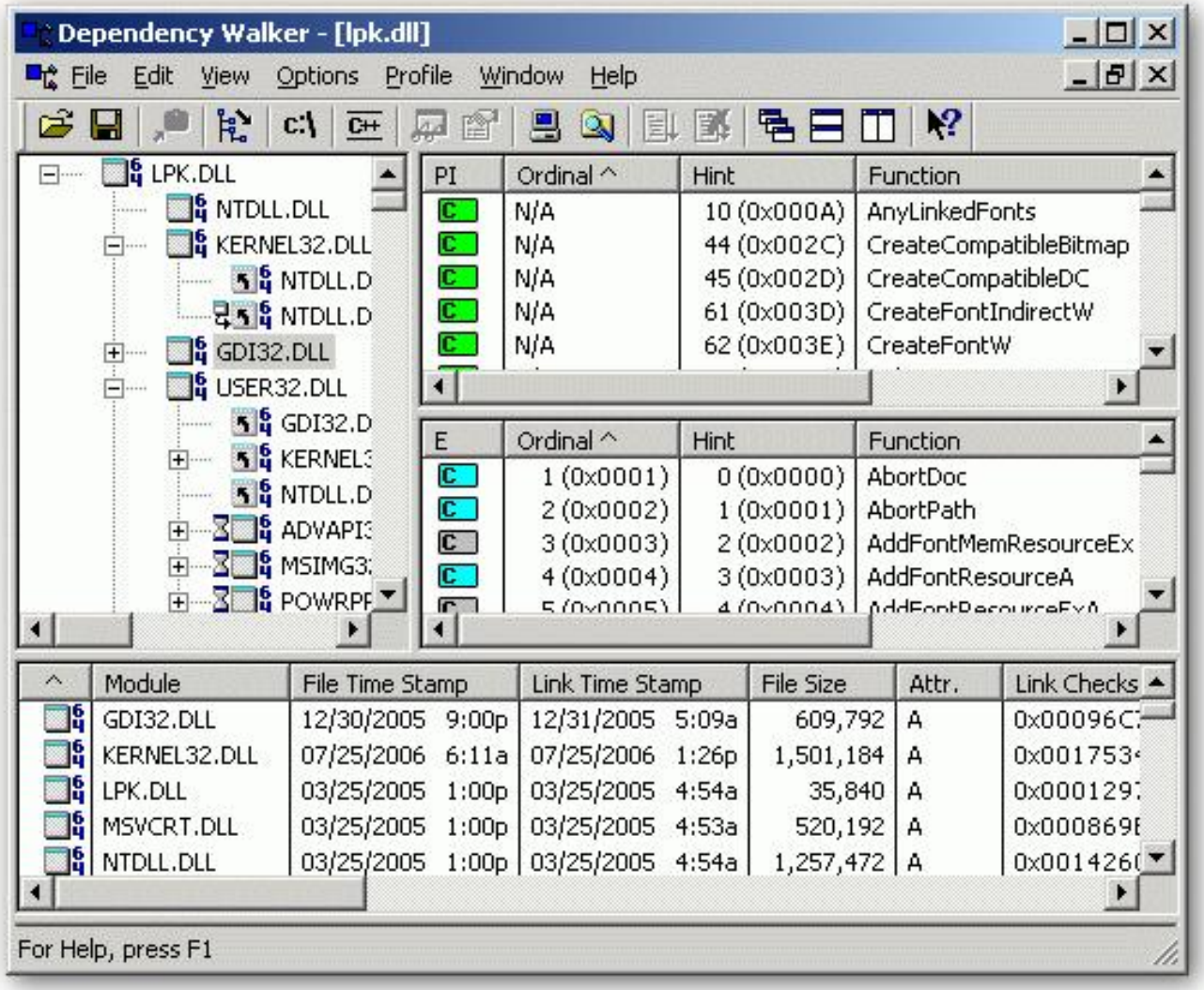
- Tentar remover o packer do malware.

## Load-time DIls ⇒ Identifica as bibliotecas carregadas no início do programa

**Dependency Walker**

- Presença de bibliotecas suspeitas ou falsas, usar Sigcheck.

# Dependency Walker



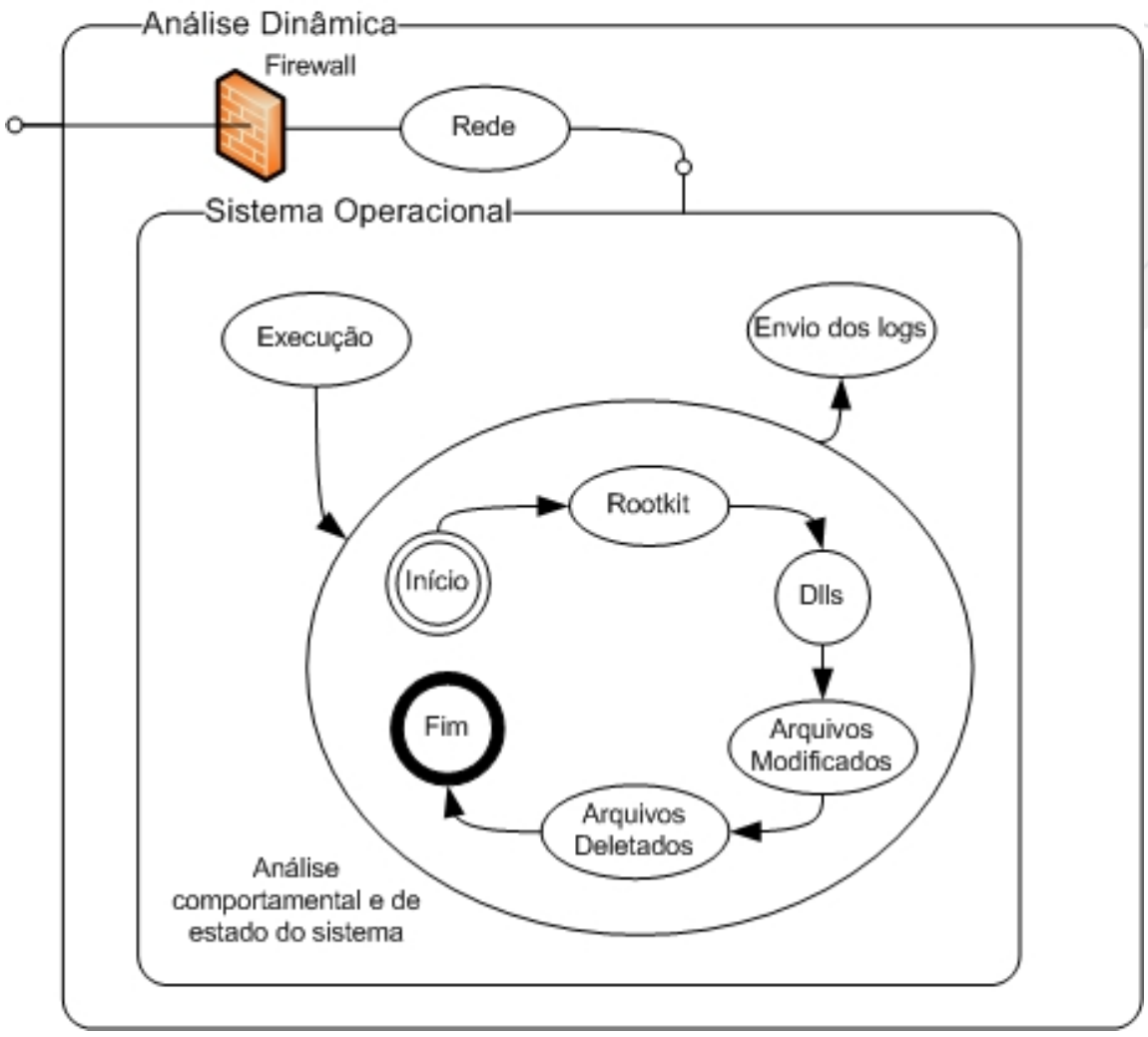

Centro de Tecnologia da Informação Renato Archer

# Cronograma



Centro de Tecnologia da Informação Renato Archer

# Ciclo da análise dinâmica

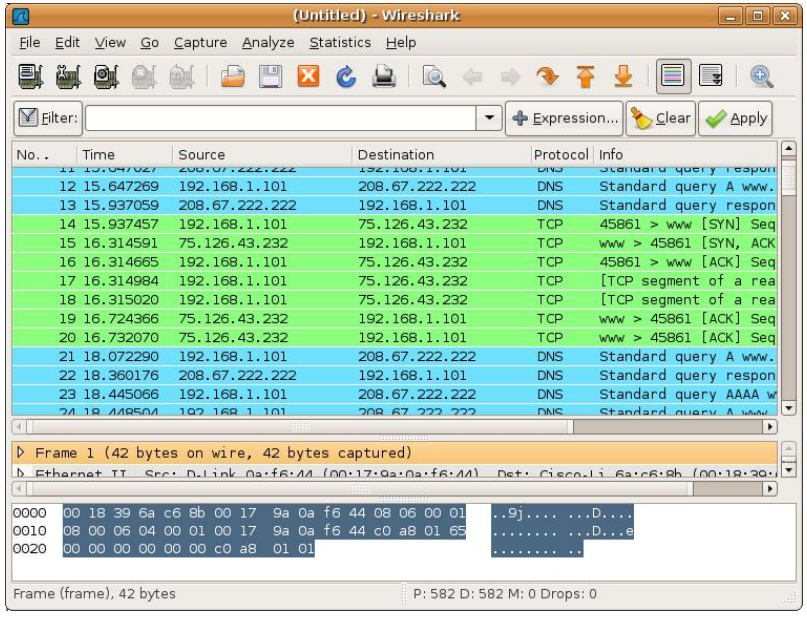



Centro de Tecnologia da Informação Renato Archer

## Monitoração do tráfego de rede

**Wireshark** - conhecer os endereços que o malware interage.

- Servidores de malwares (downloaders).
  - Bloqueio no firewall.



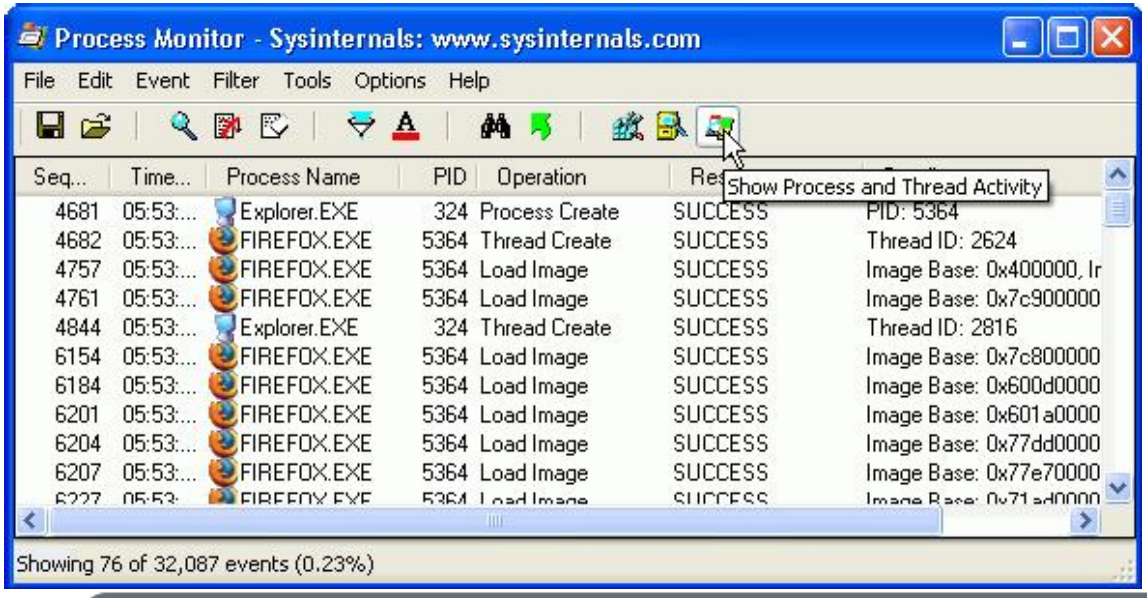

Centro de Tecnologia da Informação Renato Archer

# Análise do comportamento e estado do S.O.

## Comportamento e estado do sistema operacional

**Process Monitor** - Visualizar as chamadas de sistema que estão sendo feitas pelo malware.

- Possui filtros para refinar a monitoração;
- Indica caminhos na engenharia reversa.




Centro de Tecnologia da Informação Renato Archer

# Análise do comportamento e estado do S.O.

## Não foi suficiente?

Capture-BAT!

## Capture-BAT

Descobrir em nível de kernel quais são as ações do malware e suas modificações no sistema operacional.

- Possui filtros poderosos;
- Ajuda a descobrir a finalidade do malware;
- Indica caminhos na engenharia reversa.

# Análise do comportamento e estado do S.O.

## Não foi suficiente?

Capture-BAT!

## Capture-BAT

Descobrir em nível de kernel quais são as ações do malware e suas modificações no sistema operacional.

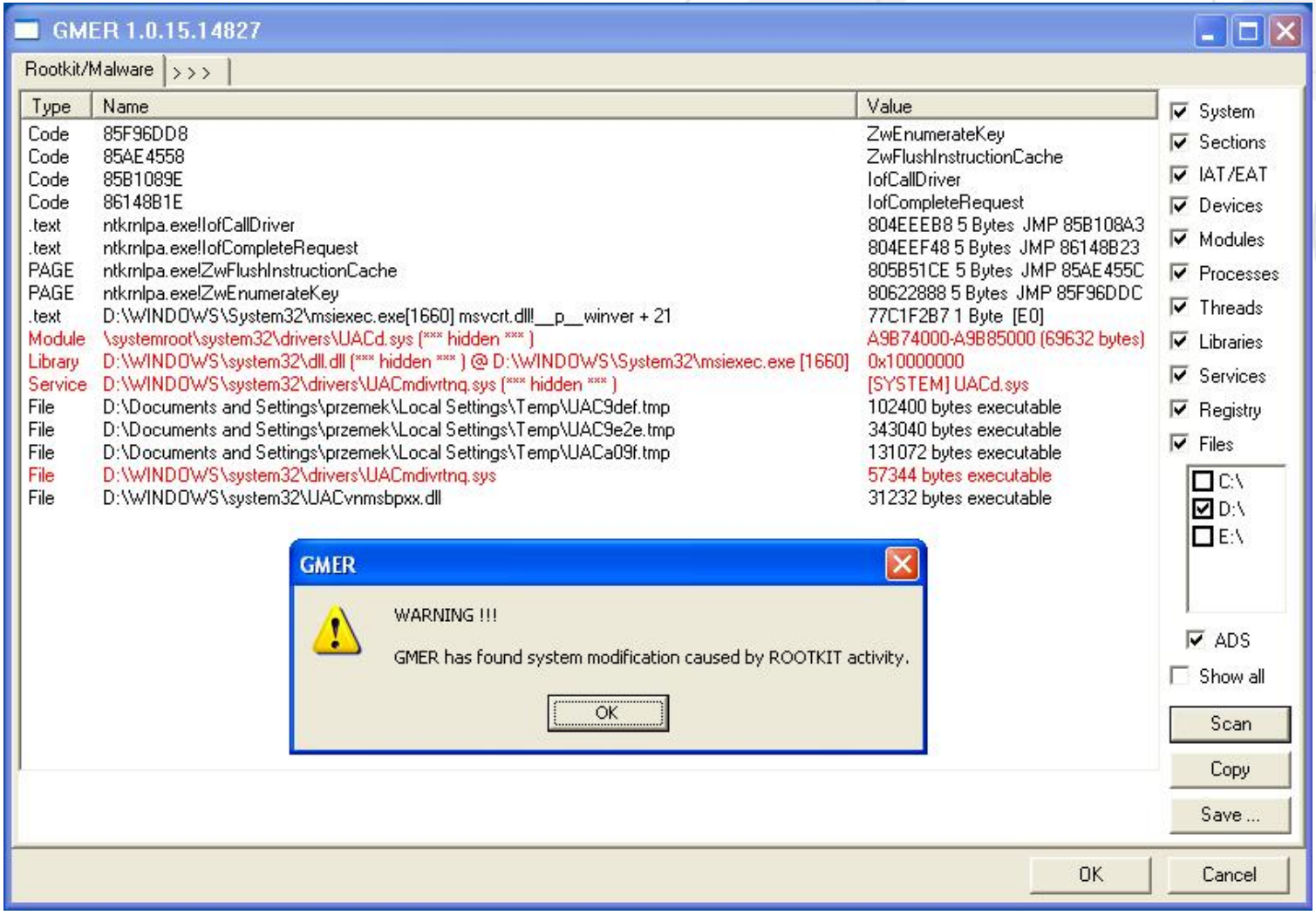
- Possui filtros poderosos;
- Ajuda a descobrir a finalidade do malware;
- Indica caminhos na engenharia reversa.

## Verificar a presença de rootkits

**Gmer** - verifica se há arquivos, registros, serviços, bibliotecas ou drivers ocultos.

- Como o próprio autor menciona, a ferramenta pode falhar;
- Verificar se os logs da rede, Process Monitor e/ou Capture-BAT condizem com o resultado.

# Gmer




Centro de Tecnologia da Informação Renato Archer

# Dlls

## Run-time Dlls

**ListDLLs v2.25** - checar se alguma biblioteca diferente das Load-time Dlls foi carregada pelo malware.

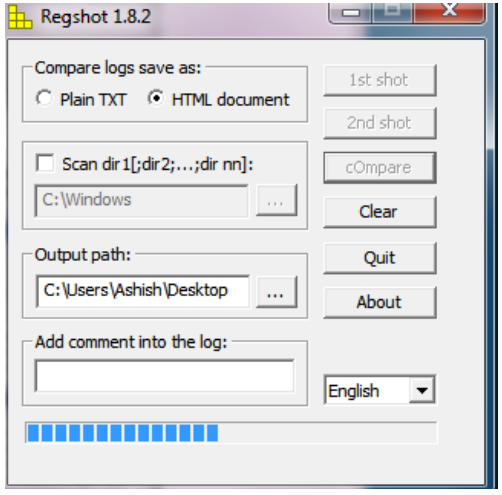
- Verificar se a biblioteca em questão está assinada (Sigcheck).

# Arquivos modificados e deletados

## Regshot

Deve-se tirar um shot do sistema intacto e outro depois do comprometimento. A ferramenta compara as duas imagens e informa as modificações.

- É possível checar por mudanças nos arquivos e nos registros;
- Há uma flag no Capture-BAT que grava os arquivos deletados.



# Cronograma

# Dump de memória

Os resultados gerados ainda não foram suficientes?

Pode-se tirar o dump da memória e analisa-la com o **Memoryze**!

## Memoryze

- Mostra todas as strings, bibliotecas, stacks e heaps dos processos que estiverem na memória;
- Lista todas as portas abertas;
- Identifica hooks, rootkits e mutex;
- Informa os módulos de kernel carregados.

## Inconveniente

Leva um longo tempo para analisar um dump de memória.

# Dump de memória

Os resultados gerados ainda não foram suficientes?
Pode-se tirar o dump da memória e analisa-la com o **Memoryze**!

## Memoryze

- Mostra todas as strings, bibliotecas, stacks e heaps dos processos que estiverem na memória;
- Lista todas as portas abertas;
- Identifica hooks, rootkits e mutex;
- Informa os módulos de kernel carregados.

## Inconveniente

Leva um longo tempo para analisar um dump de memória.

# Dump de memória

Os resultados gerados ainda não foram suficientes?
Pode-se tirar o dump da memória e analisa-la com o **Memoryze**!

## Memoryze

- Mostra todas as strings, bibliotecas, stacks e heaps dos processos que estiverem na memória;
- Lista todas as portas abertas;
- Identifica hooks, rootkits e mutex;
- Informa os módulos de kernel carregados.

## Inconveniente

Leva um longo tempo para analisar um dump de memória.

# Cronograma

# Investimento

## Gastos financeiros

- Desktop - Core 2 Duo - 2 Gb de Ram;
- HD externo;
- R$3000,00 + Profissional especializado.

## Tempo na montagem

Cerca de 1 a 2 semanas é mais que suficiente.

# Análise

## Tempo gasto

Uma análise feita na Sandbox proposta, dependendo das ferramentas utilizadas, pode levar de 30 min a 1 hora. Todavia a qualidade da análise é superior comparado aos relatórios das Sandbox existentes.

## Máquina virtual vs. real

A Sandbox foi testada em máquina virtual, por questões financeiras e de praticidade na restauração do ambiente, mas poderia ser aplicado em máquinas reais, se houver maior investimento.

# Avaliação da Sandbox

## Homemade Sandbox

- Resultados são totalmente personalizáveis as mais diversas situações;
- Apesar do maior tempo gasto para produzir um relatório, a análise é profunda e dá mais informações.

# Cronograma

# Trabalhos Futuros

Planos:

- Implementar métodos que previnam detecção de ambientes virtuais;
- Implementar o ambiente em máquinas reais;
- Pesquisar mais ferramentas para serem usadas na análise;
- Cruzar dados da análise estática com a dinâmica automaticamente;
- Armazenar informações em um Banco de Dados;

# Considerações finais

## Dúvidas?

furuse@gmail.com
victor.martins@dssi.cti.gov.br

# Considerações finais

Obrigado pela atenção!