MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m19/2010/10.05.14.35-RPQ

IDENTIFICAÇÃO DE INTRUSÃO POR ANOMALIA EM HOST COM O SISTEMA OPERACIONAL WINDOWS™ USANDO O PROCESSO DE MINERAÇÃO DE DADOS

Rogério Winter

Relatório final da disciplina Princípios e Aplicações de Mineração de Dados (CAP-359) do Programa de Pós-Graduação em Computação Aplicada, ministrada pelo professor Rafael Santos.

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/38CDELE >

INPE
São José dos Campos
2010

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@sid.inpe.br

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m19/2010/10.05.14.35-RPQ

IDENTIFICAÇÃO DE INTRUSÃO POR ANOMALIA EM HOST COM O SISTEMA OPERACIONAL WINDOWS™ USANDO O PROCESSO DE MINERAÇÃO DE DADOS

Rogério Winter

Relatório final da disciplina Princípios e Aplicações de Mineração de Dados (CAP-359) do Programa de Pós-Graduação em Computação Aplicada, ministrada pelo professor Rafael Santos.

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/38CDELE >

INPE
São José dos Campos
2010

RESUMO

Com o crescimento significativo dos sistemas computacionais em rede, principalmente os conectados à Internet, é possível constatar um crescimento exponencial das notificações de incidentes de segurança nos últimos 10 anos. Neste contexto, o presente relatório propõe um método que incorpora características de análise dinâmica de *malware* com a coleta de parâmetros de utilização de computadores para identificar intrusão por anomalia em *host*. Associado ao método é utilizado o processo de mineração de dados com algoritmos classificadores. Experimentalmente o método proposto apresentou resultados superiores quando comparado a 19 antivírus presentes no mercado de segurança da informação.

LISTA DE FIGURAS

Pág.

LISTA DE TABELAS

Pág

SUMÁRIO

1. INTRODUÇÃO

Com o crescimento significativo dos sistemas computacionais em rede, principalmente os conectados à Internet, é possível constatar pelo sítio do Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.BR, 2010), um crescimento exponencial das notificações dos incidentes de segurança nos últimos 10 anos (figura 1). De um modo geral, estes incidentes são fraudes, propagação de códigos maliciosos (*malware*- acrônimo inglês de *malicious* software*)*, ataques e invasões a computadores e suas redes. Uma associação de bancos do Reino Unido (OECD, 2010) declarou que o prejuízo direto causados por *malware* para os membros das suas organizações, em Libra Esterlina, foi de 12,2 milhões em 2004, 23,2 milhões em 2005 e de 33,5 milhões em 2006, um aumento de 90% entre 2004 e 44% de 2005. Por outro lado, é importante notar que estas perdas diretas não são totalmente representativas do impacto financeiro real, pois contribui para diminuir a confiança do cliente nas transações online, leva à perda de reputação, o impacto sobre a marca e outros custos indiretos e de oportunidade que são difíceis de quantificar.

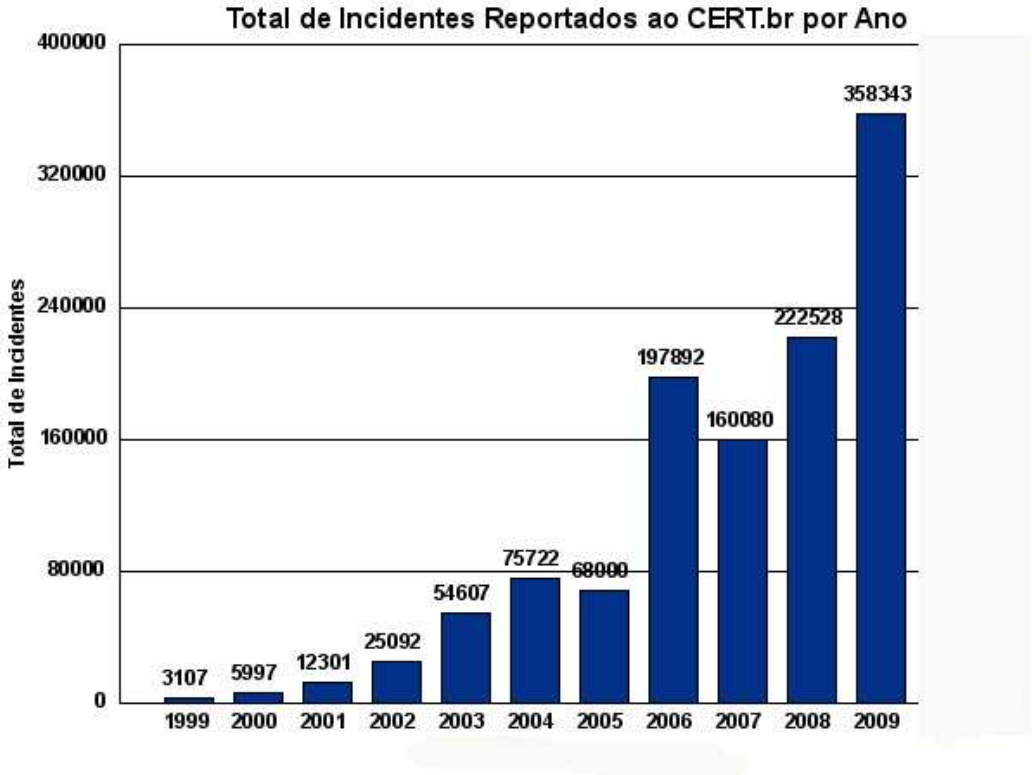

Figura 1. Histórico do total de incidentes reportados ao CERT de 1999 a 2009

Embora o uso de software livre e software *open source* tenham uma grande divulgação no meio acadêmico e de servidores, o sistema operacional Windows domina o mercado. Segundo estatísticas do (W3COUNTER, 2010) e (NETMARKETSHARE, 2010) referentes ao mês maio de 2010, tabela 01, o sistema operacional Windows é o mais usado no mundo do mercado de desktops. Além disso, segundo o sítio da (NETCRAFT, 2010), tabela 02, a Microsoft ocupa a segunda posição no mercado de servidores para a Internet.

Assim, a Microsoft possui ampla abrangência em órgãos de governo e empresas que utilizam o sistema operacional Windows para a realização das suas atividades produtivas e, inclusive, como controladores de operações de tempo real. As aplicações militares são dependentes cada vez mais de computadores e informações e é comum a utilização do sistema operacional Windows como suporte aos sistemas de tomada de decisão e sistemas de comando e controle.

Com base nas informações apresentadas, podemos inferir que a maior quantidade de incidentes de segurança como intrusões, infecções através de *malwares* e roubo de dados ocorre, na sua maioria, no sistema operacional Windows. Para (DENNING, 1987), a maior parte dos sistemas existentes têm falhas de segurança que os tornam suscetíveis a intrusões, penetrações, e outras formas de abuso; todavia, a correção de todas estas deficiências não é viável por razões técnicas e econômicas.

Na literatura, uma intrusão é definida como qualquer conjunto de ações que tentam comprometer os aspectos basilares da segurança da informação: integridade, confidencialidade e disponibilidade de um recurso computacional (CROSBIE, 2010).

Os sistemas de detecção de intrusão baseados em modelo de intrusões são classificados em detecção de intrusão por uso incorreto e detecção de intrusão por anomalia. Todos com as suas características e tecnologias empregadas, porém apresentam o inconveniente de detectarem invasões onde, muitas vezes, existe uma variação no tráfego normal, conhecidos como alarmes falsos.

A contribuição é a utilização de um método para identificar intrusão por anomalia em um sistema baseado em *host* através da utilização de processo comparativo, no qual um sistema normal é confrontado com outro invadido.

Neste contexto, são construídos os perfis de utilização com o uso da monitoração de diversas características essenciais do sistema. Por conseguinte, na experimentação, o método é aplicado em conjunto com o processo de mineração de dados (FAYYAD, 1997) com o uso do algoritmo classificador LibSVM (WITTEN, 2005). O método mostrou-se eficiente, pois na experimentação realizada comparamos o resultado com 19 antivírus conhecidos no mercado de segurança e o método apresentou a melhor precisão com o menor número de falsos negativos.

O texto está organizado da seguinte forma: a introdução que dá uma visão geral e salienta a finalidade do trabalho; revisão da literatura; na seção Ferramentas de Software e Método de Medida dos parâmetros é apresentada uma seleção dos softwares utilizados. Ainda, é definido o método, utilizado como base para a obtenção de parâmetros do computador. Em seguida, discorre-se sobre a análise preliminar dos dados com o processo de pré-processamento e transformação dos mesmos. Por fim, é demonstrada uma experimentação com aplicação do algoritmo classificador no conjunto de dados obtidos com a aplicação do método proposto, bem como a sua eficiência para o processo de detecção de intrusão por anomalia. No anexo, são apresentadas as classificações dos diversos malware utilizados no trabalho.

Tabela 1. Distribuição do Mercado de Desktop entre os sistemas operacionais.

Sistema operacional	Participação no mercado	Sistema operacional	Participação no mercado
Windows	91.62%	Windows Mobile	0.06%
Mac	5.21%	iPad	0.06%
Linux	1.05%	BlackBerry	0.05%
Java ME	0.70%	Playstation	0.04%
iPhone	0.54%	FreeBSD	0.01%
Symbian	0.24%	Nintendo Wii	0.01%
iPod Touch	0.12%	SunOS	0.01%
Android	0.09%		

Tabela 2. Distribuição do Mercado mundial de sistema operacional para servidores.

Desenvolvedor	Maio 2010	Porcentagem	Junho 2010	Porcentagem
Apache	46.608.654	55.36%	47.215.212	54.37%
Microsoft	14.977.560	17.79%	15.821.508	18.22%
Google	10.064.872	11.95%	12.282.054	14.14%
nginx	7.387.460	8.77%	6.650.907	7.66%
lighttpd	339.862	0.40%	345.251	0.40%

2. Revisão da literatura

O trabalho de (SAMI, 2010) é baseado em uma grande coleção de arquivos executáveis e abrange uma ampla gama de tipos de softwares e *malware.* O resultado foi à extração de mais de 44 mil diferentes chamadas de API importadas por 34820 arquivos PE de 890 diferentes DLLs. Os autores introduzem novas características de detecção de *malware*, as categorias chamadas API. Os dados utilizados foram obtidos de softwares disponíveis publicamente para fazer o primeiro conjunto de dados acadêmico que se destina à investigação de detecção de *malware*.

Um trabalho recente de (Y. YE, 2008), um sistema chamado IMDS, é o primeiro a tentar usar chamadas API. Eles utilizaram técnicas de mineração de dados e recursos gerados a partir de chamadas API. O conjunto de dados é composto por 12.214 programas executáveis e 17.366 *malware*. Eles usaram cerca de 2.000 arquivos PE, selecionados aleatoriamente a partir de dados de treinamento. Embora as experiências apresentassem bons resultados, devido à natureza da pesquisa industrial não há acesso aos seus dados.

Contudo, os dois trabalhos seguem o método de análise estática de *malware,* pois o estudo é realizado sem a necessidade de infecção da máquina. Assim, a análise estática (WILLEMS, 2007) uma amostra do código é examinada com o uso de um software *disassembler* ou um *decompiling* de arquivo binário.

3. Ferramentas de Software e Método de Medida dos parâmetros do computador

3.1. Software Process Monitor

Para efeito de monitoração da máquina foi utilizado o Process Monitor (RUSSINOVICH, 2010). Ele é uma ferramenta de monitoramento avançado para Windows que mostra o sistema de arquivos em tempo real, Registro e atividade de processo e thread. O Process Monitor combina os recursos de dois utilitários herdados da *Sysinternals:* o Filemon e Regmon.

Contudo, o Process Monitor acrescenta uma extensa relação de acessórios que inclui filtragem não-destrutiva e uma lista abrangente das propriedades dos eventos, tais como: IDs de sessão, nomes de usuário, informações do processo, pilhas de thread com suporte de cada operação e registro simultâneo em um arquivo. Seus recursos permitem que o Process Monitor possa ser um utilitário básico na solução de problemas do sistema. Além de estudar do comportamento da máquina para identificar ameaças e malware. Esta ferramenta é implementada como *drivers* do sistema operacional e intercepta chamadas nativas do sistema Windows (BAYER, 2006). Como resultado, ela é invisível para a aplicação que está sendo analisada. Para a execução das medidas no computador foi utilizado o Process Monitor v.2.9.

3.2. Software WEKA

Weka (FRANK, 2010) é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamados a partir de seu código Java. Weka contém ferramentas para pré-processamento, classificação, regressão, *clustering*, regras de associação e visualização dos dados. Também é adequado para o desenvolvimento de sistemas de aprendizagem de máquina. A versão utilizada na análise do conjunto de dados é a 3.7.1.

3.3. MySQL 5.1.42

O programa (SUN MICROSYSTEMS, 2010) é um servidor robusto de bancos de dados SQL (Structured Query Language – Linguagem Estruturada para Pesquisas) muito rápido, multi-tarefa e multi-usuário. O Servidor MySQL pode ser usado em sistemas de produção com alta carga e missão crítica, bem

como pode ser embutido em programa de uso em massa. O programa MySQL é de Licença Dupla. Os usuários podem escolher entre usar o programa MySQL como um produto Open Source/Free Software sob os termos da GNU General Public License[1] ou podem comprar uma licença comercial padrão da MySQL.

3.4. Visulab

(VISULAB, 2010) é abreviação de laboratório de visualização e é um pacote de software experimental para a visualização comparativa dos dados multivariados.

Neste contexto a palavra comparativa define a possibilidade de mostrar os dados multivariados em janelas diferentes simultaneamente, utilizando assim métodos de visualização diferentes. Por "dados multivariados" entendemos dados multidimensionais com muitas variáveis coordenadas. A mais recente versão deste software é concebida como um *add-in* para trabalhar no Microsoft Excel 2000, XP, 2003 e acima. *VisuLab* é uma marca registrada da ETH Zurich.

3.5. Método de medida para aquisição dos parâmetros do computador

O método utilizado neste trabalho permitiu estudar os fenômenos que distinguem o comportamento de um computador em atividade normal de outro com problemas com infecção ou intrusão. Uma característica do método proposto é derivada da análise dinâmica de *malware (WILLEMS, 2007)*, onde o comportamento do *malware* é analisado acompanhando-se todos os registros das operações no nível do sistema por certo tempo. Outro requisito fundamental para o exercício da análise é que a fase de infecção seja conduzida em um ambiente isolado, a exemplo de máquinas virtuais.

Em função das observações realizadas no comportamento de diversos *malware* constatou-se que o tempo para o início das atividades maliciosas dos mesmos era protelado por alguns minutos. Assim, definiu-se o intervalo de medição, em cada fase, entre cinco minutos e dez minutos, exceto para a fase de instalação, em razão da velocidade do computador, os tempos ficaram entre um minuto e cinco minutos.

[1] http://www.fsf.org/licenses/)

Assim, para a efetivação das medidas, o método foi concebido em fases onde foram estabelecidos cinco momentos de utilização de um computador: fase inicial; fase de instalação; fase de conexão de rede; fase de operação; e fase de infecção do sistema subdividida em subfases para medir cada tipo de *malware*.

* Fase inicial - é caracterizada por um processo de pós-instalação sem a aplicação de pacotes de atualização ou de segurança. São realizadas atividades com a máquina com os programas instalados por padrão do sistema operacional.
* Fase de instalação - são realizadas instalações de software consideradas úteis para a utilização do computador e adaptadas as necessidades de trabalho. Na seção da experimentação são apresentados os diversos softwares instalados.
* Fase de conexão de rede - o computador é conectado à rede local e na Internet e assim são realizados comandos de rede para ajustes dos parâmetros de conexão. Estes comandos são específicos do administrador da máquina.
* Fase de operação - caracterizada pela operação propriamente dita do sistema, onde os softwares instalados foram explorados nas tarefas específicas.
* Fase de infecção - dividida em diversas subfases para avaliar os vários tipos de infecção e apreciar as diferentes operações realizadas pelo computador. A máquina de teste é infectada com *malware* para que se pudessem realizar as medidas dos parâmetros de utilização necessários ao estabelecimento do perfil da máquina.

4. Experimentação

Toda a experimentação foi conduzida em um ambiente de máquinas virtuais (VMWARE, 2010), onde cada fase foi medida separadamente na máquina virtual conforme descrito na seção 3.2 deste artigo e visualizado na figura 1.

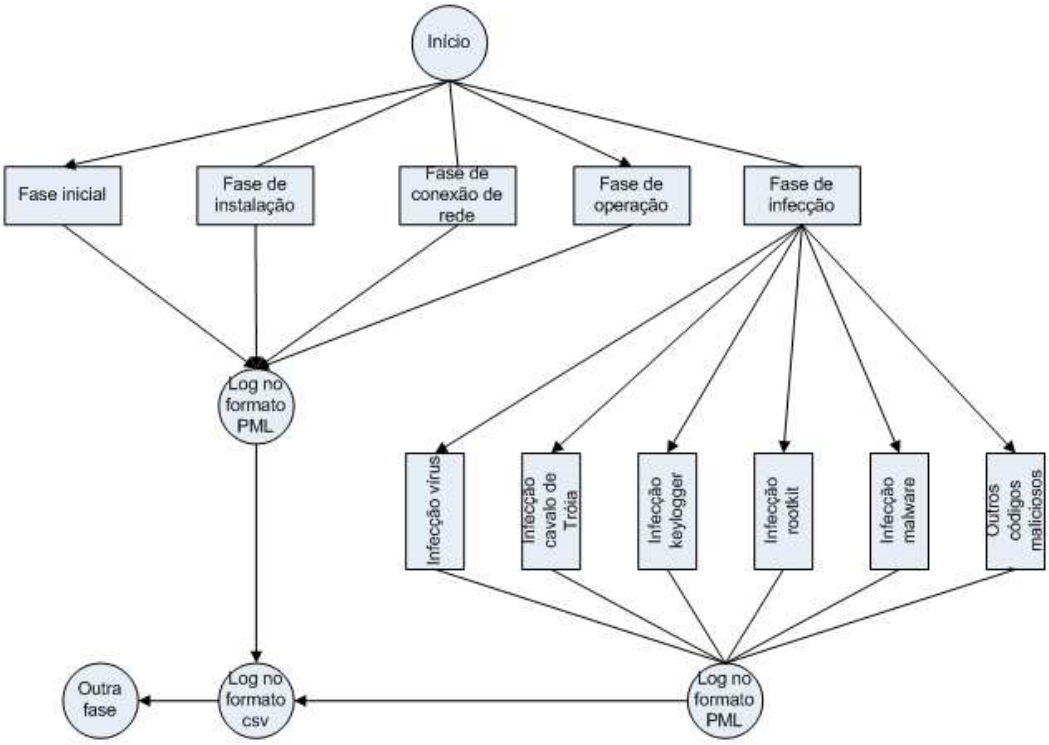

Figura 2 Método de medidas do computador

As informações obtidas através da monitoração do computador com o software Process Monitor permitiram a constituição de uma base de dados de *logs* com os diversos parâmetros de operação. O Process Monitor realiza a monitoração em cinco classes distintas: *Registry Activity, File System Activity, Network Activity, Process and Thread Activity* e *Profiling Events*. Para efeito deste trabalho utilizamos as seguintes classes:

- *Registry* – monitora as operações realizadas no registro;
- *File System* – monitora as operações no sistema de arquivos do Windows;
- *Network* – realiza a monitoração das atividades de rede incluindo protocolos;
- *Process and Thread* – monitoração dos processos e threads inicializados pelos mesmos;

Além disso, para efeito deste trabalho foram utilizadas as seguintes opções de monitoração: Time of Day, Relative Time, Duration, Process Name, PID, Event Class, Operation, Result, Image Path, User, TID, Path e Parent ID.

- Time of Day - o parâmetro marca a hora do dia na qual foi executada determinada operação no sistema.

- Relative Time - o tempo de operação em relação ao início do processo do Monitor ou a última vez que o processo de visualização do monitor foi cancelado.
- Duration - duração de uma operação que foi concluída.
- Process Name - o nome do processo em que ocorreu o evento.
- PID - o Process ID (PID) que executou uma determinada operação.
- Event Class - a classe (File, Registry, Process) do evento.
- Operation - a específica operação de um evento. (Exemplo Read, RegQueryValue, etc.).
- Result - o código de status de uma operação concluída.
- Image Path - o caminho completo da imagem em que o processo está rodando.
- User- o nome da conta do usuário na qual o processo executou uma operação.
- TID - O Thread ID (TID) que executa uma operação.
- Path - o caminho de um recurso que um evento referencia.
- Parent ID - a referência ao processo pai que executou uma operação.

Com a finalidade de traçar o perfil de utilização do computador foram identificados alguns parâmetros mais relevantes para distinção entre atividade “normal” e “intrusão”. Em seguida, estes parâmetros são conceituados.

Fast I/O (RUSSINOVICH, 2009) é um tipo de operação projetada especificamente para uma rápida sincronização I/O no cache de arquivos. Em operações *Fast I/O*, os dados são transferidos diretamente entre buffers de usuário e o sistema de cache, ignorando o sistema de arquivos e o controlador de armazenamento pilha.

I/O request packets (IRPs) (RUSSINOVICH, 2009) são estruturas do modo *kernel* usadas pelo *Windows Driver Model* (WDM) e drivers de dispositivo do Windows NT para se comunicarem uns com os outros e com o sistema operacional. Eles são estruturas de dados que descrevem as solicitações de I/O.

User Datagram Protocol (UDP) (STEVENS, 1997) e *Transfer Control Protocol* (TCP) são os dois protocolos da camada de transporte do TCP/IP nas versões 4 e 6. O UDP é um protocolo orientado a datagrama; sem um

mecanismo de confiança que garanta a entrega do datagrama ao destinatário e sem conexão, pois não necessita manter um relacionamento entre cliente e servidor. Alguns protocolos que utilizam o UDP são: *Dynamic Host Control Protocol* (DHCP), *Domain Name Service* (DNS), *Trivial File Transfer Protocol* (TFTP) e *Simple Network Management Protocol* (SNMP).

O Windows (HONEYCUTT, 2005) armazena dados de configuração no registro. O registro é um banco de dados hierárquico, que pode ser descrito como um repositório central de informações de configuração, na terminologia da Microsoft. Isso permite que as configurações sejam referenciadas utilizando caminhos semelhantes aos caminhos de arquivo no Windows Explorer. Diversos tipos de operações podem ser executadas no registro: consulta de parâmetros, exclusão de parâmetros, atualização de parâmetros e outros.

Em todas as fases e subfases os resultados das medidas, os *logs*, foram salvos em arquivos com o formato c*omma-separated values* (csv), em virtude da facilidade de transportá-los para o banco de dados. A fim de constituir uma base de dados criada no MySQL, foi desenvolvido um aplicativo em linguagem *shell script*, no Linux, que lê os diferentes arquivos no formato *csv*, ajusta determinados campos e armazena as informações para posterior sumarização.

Para a fase de instalação os softwares constantes da tabela 3 foram selecionados como adequados a experimentação do método em virtude da sua ampla divulgação e pela facilidade de obtenção na Internet, além de poderem ser instalados sem a aquisição de licença.

Tabela 3. Aplicativos utilizados nos testes.

Aplicativo	Tipo	Versão
Br Office	Escritório	3.1.1
Adobe Acrobat Reader	Leitor de arquivos no formato PDF	9.3
Winrar	Compactador de arquivos	3.9.1
Java Runtime Environment (JRE)	Máquina virtual Java	6 Update 20
Firefox	Navegador de páginas Web	3.5.7
Kaspersky 2010	Antivírus	9.0.0.736

Na fase de conexão de rede o computador foi conectado fisicamente a rede local para a realização das medidas dos parâmetros de operação. Para isso, um arquivo batch com os comandos específicos de ajuste das conexões de rede foi utilizado para automatizar a operação.

A fase de infecção a máquina foi infectada com 41 tipos de *malware* (anexo 1) obtidos em **(VX HEAVENS, 2010)**. Todos os softwares maliciosos foram analisados antecipadamente pelo verificador de *malware* do Jotti **(JOTTI, 2010)**. Este analisador é um serviço online gratuito que permite a análise de arquivos suspeitos com 19 antivírus conhecidos no mercado de segurança.

Em virtude da tecnologia empregada em cada antivírus, os *malware* foram classificados diferentemente. Entretanto, alguns antivírus apresentaram falsos negativos **(CROSBIE, 2010)**, ou seja, não identificaram ameaça onde na verdade existia o *malware* (figura 1). Desta forma, o resultado da análise constituiu a informação quanto à habilidade do software antivírus em detectar uma ameaça ao sistema computacional. O objetivo é determinar se o computador está comprometido ou não e comparar a porcentagem de falso negativo com o método (figura 2).

Antivírus	ArcaVir	Avast	AVG	AntiVir	bitdefender	ClamAV	Cpsecure	Dr.WEB	F-PROT	F-Secure	G DATA	IKARUS
indicativo	A	B	C	D	E	F	G	H	I	J	K	L
porcentagem falso negativo	31.71	73.17	12.20	4.88	12.20	24.39	43.90	14.63	12.20	14.63	9.76	4.88

Antivírus	Kaspersky	NOD 32	PANDA	Quick Heal	SOPHOS	VBA 32	VirusBuster
indicativo	M	N	O	P	Q	R	S
porcentagem falso negativo	12.20	7.32	65.85	34.15	14.63	29.27	26.83

Figura 3. Comparativo dos antivírus

5. Pré-processamento e seleção dos parâmetros

Na sequência, foi realizada a consulta ao banco de dados para cada fase e subfase com a finalidade de se estabelecer um perfil comportamental da máquina, discriminando um comportamento normal do infectado ou invadido.

Através da análise visual dos dados, utilizando o software (VISULAB, 2010), foi possível selecionar alguns atributos mais relevantes ou os que apresentavam um maior impacto na operação do sistema. Entretanto, em

virtude da quantidade de parâmetros, 83 no total, a seleção dos mesmos foi prejudicada.

Assim os dados, ainda no formato *csv*, foram carregados no software Weka. A fim de efetuar uma seleção dos dados mais relevantes, no ambiente do software Weka os mesmos foram avaliados pelo algoritmo *InfoGainAttributeEval* (WITTEN, 2005), que é um método de avaliação para a seleção de atributos, no qual mede-se o ganho de informação de acordo com a classe do atributo. Este método pode tratar um valor ausente como um valor separado ou distribuir as contagens entre outros valores, proporcionalmente à sua freqüência. Desta forma, o algoritmo apontou aqueles parâmetros mais significativos no conjunto de dados, os demais considerados de pouca expressão foram retirados. Finalmente, o conjunto de dados ficou com 782 instâncias e 62 atributos.

6. Seleção e aplicação do algoritmo

A detecção de intrusão por anomalia é um problema tipicamente de classificação, pois desejamos distinguir um computador em um estado normal de outro infectado ou invadido. Particularmente na experimentação foi utilizada a classificação supervisionada.

A etapa seguinte o conjunto de dados foi submetido ao algoritmo de mineração de dados do Weka (FRANK, 2010), o LibSVM (CHANG, 2001). O *Suport Vector Machine* (SVM) é uma popular técnica de classificação. O algoritmo foi utilizados com os parâmetros padrão oferecidos no Weka.

Para efeito da mineração de dados com o algoritmo classificador, os mesmos foram submetidos à opção de teste do Weka (BOUCKAERT, 2010) *cross validation* ou validação cruzada. Na modalidade validação cruzada e a fim de obter resultados estatisticamente mais significativos, utilizou-se 10 iterações. Neste caso, a validação cruzada é realizada 10 vezes em cada *fold*, isto significa que são realizadas 100 chamadas de um classificador nos dados para treinamento e novamente testadas com dados de teste.

O algoritmo classificador apresentou os resultados conforme as figuras 4.

=== Summary ===			
Correctly Classified Instances	703	89.8977	%
Incorrectly Classified Instances	79	10.1023	%
Kappa statistic	0.3953		
Total Number of Instances	782		

Figura 4. Sumário do algoritmo LibSVM.

Detalhe de precisão por classe	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.275	0	1	0.275	0.432	0.638	normal
	1	0.725	0.895	1	0.945	0.638	infeccao
Weighted Avg.	0.899	0.624	0.91	0.899	0.873	0.638	

Figura 5. Detalhe de precisão do algoritmo LibSVM.

=== Confusion Matrix ===			
a	b	<-- classified as	
30	79	|	a = normal
0	673	|	b = infeccao

Figura 6. Matriz de confusão do algoritmo LibSVM.

7. Análise dos resultados

Como resultado do tratamento nos dados obtidos pela monitoração do computador de teste nas diversas fases do método proposto na seção 3.2, os mesmos foram submetidos à análise do algoritmo classificador LibSVM. O algoritmo analisou 782 instâncias amostradas cada uma com 62 atributos pertinentes a parâmetros de utilização do computador de teste.

O classificador foi avaliado utilizando a validação cruzada com 10 dobras, o que nos permitiu utilizar os dados como um todo e os dados de treinamento para tornar os resultados dos testes mais confiáveis.

Os dados possuem uma natureza desbalanceada, pois temos mais instâncias rotuladas como infecção do que como normal. Assim, utilizando os resultados de precisão tradicional avalia-se que essa classificação não é adequada. Algumas técnicas foram introduzidas por (BATISTA, 2004) para fazer uma avaliação mais confiável quando se lida com conjuntos de dados desequilibrada.

Foi escolhida a área sob a curva ROC como medida de precisão para avaliar a experiência, além disso, taxa de verdadeiro negativo e taxa de falso positivo para a classe de infecção.

Pela análise da área da curva ROC (*Receiver Operating Characteristic*) (tabela 4), observamos que o algoritmo LibSVM, demonstrou uma baixa precisão. O principal responsável por este resultado foi a quantidade de instâncias com o rótulo normal. O conjunto de dados submetido à classificação possui um total de 782 instâncias sendo 109 da classe normal e 673 da classe infecção.

No tocante a porcentagem de classificações corretas o algoritmo apresentou resultado em torno dos 90% o que nos dá um excelente resultado.

Ao analisarmos a taxas de verdadeiro negativo e falso negativo para a classe infecção constatamos que o desempenho do algoritmo para detectar uma infecção dos *malware* utilizados na experimentação foi excepcional. Neste aspecto, foram identificadas todas as instâncias da classe infecção.

Comparativamente, a utilização do método de medidas aliado ao processo de mineração de dados, com o algoritmo LibSVM, apresentou um desempenho superior ao dos antivírus utilizados como referência ao trabalho (figura 1).

8. Conclusão

Tradicionalmente, seria mais difícil procurar por atividade não autorizada em um sistema, pois poderiam existir tantos falsos alarmes que os mesmos se assemelhariam a atividade legítima.

Entretanto, utiliza-se um método para identificar intrusão por anomalia através da utilização de processo comparativo, no qual um sistema normal é confrontado com várias instâncias de sistemas invadidos. Para efetivação das

medidas, o método foi concebido em fases onde foram estabelecidos cinco momentos de utilização de um computador: fase inicial; fase de instalação; fase de conexão de rede; fase de operação; e fase de infecção do sistema. Assim, foi possível construir os perfis de utilização, além de definir, com o uso da monitoração, as características mais relevantes do sistema.

Com a seleção dos parâmetros do computador, a experimentação utilizou o processo de mineração de dados com a aplicação do algoritmo classificador LibSVM onde as características anteriormente selecionadas contribuíram decisivamente na detecção de intrusão por anomalia.

O trabalho como um todo conseguiu apresentar resultados excelentes na detecção de intrusão, em parte pela inserção do método que propiciou, de forma controlada, a monitoração do sistema computacional. Por outro lado, a utilização dos algoritmos fortaleceu a tese de reduzir a quantidade de falsos negativos no sistema de detecção de intrusão. Ressalte-se que o método traça um perfil do ambiente de trabalho do computador, identificando momentos de atividade normal e de intrusão e infecção. O método pode ser aplicado a outros sistemas operacionais com adaptações no software de monitoração do sistema computacional.

Por fim, o método de medida utilizado no trabalho agregado à utilização do algoritmo classificador LibSVM revelaram eficiência e precisão na detecção de intrusão por anomalia em *host* quando aplicados a base de dados em estudo. Desta forma, o seu uso em prol da segurança da informação é perfeitamente adequado como mecanismo de proteção.

Referências Bibliográficas

BATISTA, G. E.; PRATI, R. C.; A. MONARDI, M. C. A study of the behavior of several methods for balancing machine learning training data. **SIGKDD Explorations. Newsl**, v. 1, n. 6, p. 20-29, jun 2004. ISSN http://doi.acm.org/10.1145/1007730.1007735.

BAYER, U. et al. Dynamic analysis of malicious code. J**ournal in Computer Virology**, Paris, v. 2, n. 1, p. 67–77, August 2006.

BOUCKAERT, R. R. et al. WEKA Manual for Version 3-7-1. **WEKA Documentation**, 2010. Disponivel em: <http://www.cs.waikato.ac.nz/~ml/weka/>. Acesso em: 10 março 2010.

CERT.BR. Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. Estatísticas dos Incidentes Reportados ao CERT.br., 2010. Disponível em: <http://www.cert.br/stats/incidentes/>. Acesso em: 15 julho 2010.

CHANG, C.-C. LIN, C.-J. **LIBSVM** : a library for support vector machines. LIBSVM, 2001. Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/libsvm >. Acesso em: 01 julho 2010.

CROSBIE, M. P. K. E. A. **Intrusion detection pages.** 2010. Disponível em: <http://www.cerias.purdue.edu/about/history/coast_resources/idcontent/welcome.html>. Acesso em: 04 fevereiro 2009.

DENNING, D. E. An Intrusion-Detection Model. **IEEE Transactions on Software Engineering**, v. SE-13, NO. 2, p. 222-232, february 1987.

FAYYAD, U.; U T H U R U S A M Y, R. **From data mining to knowledge discovery in databases.** American Association for Artificial Intelligence, 1997. Disponível em: <http://www.aaai.org/Conferences/National/1997/aaai97.html>. Acesso em: 25 março 2010.

FRANK, E. H. M. E. A. **WEKA Version 3-7-1**. The University of Waikato, 2010. Disponível em: <http://www.cs.waikato.ac.nz/~ml/weka/>. Acesso em: 10 março 2010.

HONEYCUTT, J. **Windows registry guide**. Washington: Microsoft Press, v. 1, 2005.

JOTTI. **Verificador de malware do Jotti.** 2010. Disponível em: <http://virusscan.jotti.org/pt-br>. Acesso em: 01 julho 2010.

NETCRAFT. **July 2010 Web Server Survey**. 2010. Disponível em: <http://news.netcraft.com/>. Acesso em: 11 julho 2010.

NETMARKETSHARE. **Operating system market share**. 2010. Disponivel em: <http://www.netmarketshare.com/report.aspx?qprid=8&qpmr=1000&qptimeframe=Y&qpct=2#>. Acesso em: julho nov. 2010.

Organisation for Enonomic Co-operation and Development (OECD). **Malicious Software (Malware)**: a security threat to the Internet Economy. 01 julho 2010.

Disponível em: <http://www.oecd.org/dataoecd/53/34/40724457.pdf>. Acesso em: 01 julho 2010.

RUSSINOVICH, M. C. B. **Process monitor**. Sysinternals, 2010. Disponível em: <http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>. Acesso em: 10 abril 2010.

RUSSINOVICH, M. S. D. A. I. A. **Windows internals –** covering Windows Server 2008 and Windows Vista. 5. ed. Redmond: Microsoft Press, v. 1, 2009.

SAMI, A. E. A. Malware detection based on mining API calls. In: ACM Symposium on Applied Computing, 25., 2010, Sierre. **Proceedings…** Sierre: ACM, 2010. p. 1020-1025.

STEVENS, W. R. **TCP/IP Illustrated:** the protocols. 1. ed. Massachusetts: Addison Wesley Longman, Inc, v. 1, 1997.

SUN MICROSYSTEMS. **MYSQL versão 5.1.42**. MYSQL, 2010. Disponível em: <http://dev.mysql.com/downloads/>. Acesso em: 12 janeiro 2010.

VISULAB. **Interactive data visualisation in Microsoft Excel**. 2010. Disponível em: <http://www.inf.ethz.ch/personal/hinterbe/Visulab/>. Acesso em: 2 abril 2010.

VMWARE. Product Support for VMware Workstation. 2010. Disponível em: <http://www.vmware.com/support/product-support/workstation/>. Acesso em: 12 fevereiro 2010.

VX HEAVENS. **Computer virus collection**. 2010. Disponível em: <http://vx.netlux.org>. Acesso em: 12 abril 2010.

W3COUNTER. **Global web stats.** 2010. Disponível em: <http://www.w3counter.com/globalstats.php>. Acesso em: 11 julho 2010.

WILLEMS, C. H. F. Toward automated dynamic malware analysis using CWSandbox. **IEEE Security and Privacy**, v. 5, n. 2, p. 32-39, Mar 2007. ISSN doi:10.1109/MSP.2007.45.

WITTEN, I. H. F. E. **Data mining practical machine learning tools and techniques.** 2. ed. San Francisco: Morgan Kaufmann Publishers, 2005.

YANFANG, Y.; DINGDING, W.; TAO, L.; DONGYI, Y. An intelligent pe-malware detection system based on association mining. **Journal in Computer Virology**, France, v. 4, p. 323–334, Feb. 2008. ISSN DOI 10.1007/s11416-008-0082-4.

	Fabricantes	ArcaVir	avast!	AVG	AntiVir
Ordem	Codigo Malicioso	ArcaVir	Avast	AVG	AntiVir
1	IRC.LIDA.zip	Worm.Dos.Funex.a	Nada encontrado	IRC-Worm/Generic.L	Worm/Funex.A
2	WKILL_I.zip	Trojan.Vb.Afw	Nada encontrado	Generic.JKL	TR/Crypt.ULPM.Gen
3	SysDel.a.zip	Trojan.Dos.Delsystem.l	Nada encontrado	Generic.TUB	TR/DelSystem.L
4	SysDel.b.zip	Trojan.Dos.Delsystem.m	Nada encontrado	Generic.TUC	TR/DelSystem.M
5	Virus_Evil.rar	Nada encontrado	Nada encontrado	Win32.Generic.AP	W32/Bloodhound.A
6	W32.AmirCivi.a.zip	Worm.Agent.M	Nada encontrado	Worm/Generic.PW	Worm/P2P.Agent.m.1
7	W32.AmirCivil.b.zip	Nada encontrado	Nada encontrado	Worm/Generic.PY	Worm/P2P.Agent.m.2
8	W32.AmirCivil.c.zip	Worm.Agent.O	Nada encontrado	unknown virus Win32/DH.AA54534F48	Worm/Agent.O
9	W32.Autoexec.zip	Trojan.Vb.Alw	Nada encontrado	Generic.TLG	TR/VB.alw
12	W32.Dosman.zip	Nada encontrado	Nada encontrado	Generic15.ACTU	HEUR/Crypted
11	W32.Irvig.zip	Worm.Vb.ge	Nada encontrado	Generic5.PLK	TR/Agent.24921
10	W32.Mimi.zip	Antivírus indisponíve	Nada encontrado	Worm/VB.EB	TR/Crypt.CFI.Gen
13	W32.Natalia.a.zip	Worm.Vb.bi	Nada encontrado	Worm/VB.IZ	TR/SubGAM
14	W32.Natalia.b.zip	Nada encontrado	Nada encontrado	Generic16.GSC	TR/Scar.atwl
16	W32.Tori.zip	Worm.Vb.jt	Nada encontrado	VB.AJ	Worm/Generic.A.24
17	wiki.rar	Trojan.Agent.Un	Nada encontrado	Worm/Generic.OX	TR/Crypt.XPACK.Gen
19	XPAC.zip	Worm.Irc.Pucker.A	Nada encontrado	Worm/Generic.OT	TR/Puck.A
20	Worm.Win32.Scorvan.zip	Worm.Scorvan	Nada encontrado	Worm/Scorvan.A	Worm/Scorvan
21	W32.Neron.b.zip	Nada encontrado	Nada encontrado	Generic17.BEIQ	TR/Sisron.D
22	W32_Neron.a.zip	Trojan.Vb.Afx	Nada encontrado	Generic.JHO	TR/VB.afx
23	W32_Eliza.zip	Hll.Eliz.a	Nada encontrado	Worm/VB.JA	W32/Eliza
24	W32.Pardis.b.zip	Nada encontrado	Nada encontrado	Generic17.BEIP	TR/Sisproc.E
25	W32.Pardis.zip	HLL.Vb.X	Nada encontrado	Worm/VB.QV	W32/VB.x.1.B
26	W32.Urchin.zip	Trojan.Vb.Acd	Nada encontrado	Generic2.AVY	TR/VB.acd.1
27	Win32.Leon.rar	Nada encontrado	Nada encontrado	Nada encontrado	W32/Zazel.A
28	Win32.AnTaReS.rar	Trojan.Genome.arxe	Nada encontrado	Win32/Ares	Worm/Agent.20566.A
43	TrojanSimulator.exe	Nada encontrado	Win32:TrojanSim	Nada encontrado	Nada encontrado
29	PDM.keylogger.exe	Nada encontrado	Nada encontrado	Nada encontrado	Nada encontrado

bitdefender	ClamAV	Cpsecure	Dr.WEB
Nada encontrado	Nada encontrado	Nada encontrado	IRC.Lida
Trojan.Vb.AFW	Trojan.Generic.Bredolab-2	Troj.W32.VB.afw	Trojan.KillFiles.490
Trojan.Dos.Delsystem.L	Nada encontrado	Nada encontrado	Trojan.DelSys
Trojan.Dos.Delsystem.M	Nada encontrado	Nada encontrado	Trojan.DelSys
Win32.Bloodhound.A	Nada encontrado	Nada encontrado	modification of Win32.Iwing.3560
Win32.Worm.Amire.A	PUA.Packed.FSG	W32.P2P.W.Agent.M	Win32.HLLW.Amir
Win32.Worm.Amire.B	Trojan.Pakes-2516	Nada encontrado	Win32.HLLW.Amir
Win32.Worm.P2p.Agent.O	Trojan.Pakes-2516	W32.P2P.W.Agent.O	Win32.HLLW.Amir
Trojan.Vb.ALW	Trojan.Generic.Bredolab-2	Troj.W32.VB.alw	Trojan.Hesy
Gen:Trojan.Heur.PT.aiadaiHa9xpi	PUA.Packed.FSG	Nada encontrado	Nada encontrado
Generic.Malware.SPCTk.21E4927D	Trojan.Pakes-2516	W32.Email.W.generic	Win32.HLLM.Generic.399
Win32.Worm.VB.AQ	Trojan.Generic.Bredolab-2	W32.W.VB.av	BackDoor.Generic.1110
Win32.Worm.VB.BI	Trojan.Generic.Bredolab-2	W32.W.VB.bi	Win32.HLLW.Natalia
Trojan.Generic.3674003	Trojan.Pakes-2516	Nada encontrado	Nada encontrado
Trojan.Dropper.EX	Trojan.Small-1617	BadJoke.W32.VB.am	Joke.Forgery.141
Generic.Malware.dld!!.6490323D	W32.Wicket	Troj.W32.Agent.un	Win32.HLLW.Wiki
IRC-Worm.Pucker.A	Worm.Puker	Nada encontrado	HLLO.Cruel
Generic.Malware.N!g.F190922B	Worm.P2P.Scorvan	W32.W.Scorvan	Win32.HLLW.Vandeed.6
Trojan.Generic.3673998	PUA.Packed.FSG	Nada encontrado	Nada encontrado
Trojan.Vb.AFX	Trojan.Generic.Bredolab-2	Troj.W32.VB.afx	Trojan.Neron
Backdoor.981	Trojan.Generic.Bredolab-2	W32.VB.Z	Win32.HLLP.Eliza.12288
Trojan.Generic.3674002	PUA.Packed.FSG	Packed.W32.Krap.ag	Nada encontrado
Trojan.Vb.NBC	Nada encontrado	W32.VB.X	Win32.HLLC.Pardis
Trojan.VB.ACD	Trojan.Generic.Bredolab-2	Nada encontrado	Win32.HLLP.Urchin.6656
Trojan.Generic.580332	Virtool.Kpasm	W32.Zezal.a	Nada encontrado
Nada encontrado	Nada encontrado	Packed.W32.Tdss.c	modification of Win32.Benny.6382
Nada encontrado	Riskware.Trojansimulator	WebToolbar.W32.FenomenGame.pow	Program.AVTest
Nada encontrado	Nada encontrado	Nada encontrado	Nada encontrado

F-PROT	F-Secure	G DATA	IKARUS
Malware!04e0	IRC-Worm.DOS.Funex.a	Nada encontrado	IRC-Worm.DOS.Funex
W32/SillyTrojan.LH	Trojan.Win32.VB.afw	Trojan.Vb.AFW	Trojan.Win32.VB
Malware!2685	Trojan.DOS.DelSystem.l	Trojan.Dos.Delsystem.L	Trojan.DOS.DelSystem
Malware!7cf2	Trojan.DOS.DelSystem.m	Trojan.Dos.Delsystem.M	Trojan.DOS.DelSystem
W32/SuspPack.T.gen!Eldorado	Virus.Win32.Agent.ab	Win32.Bloodhound.A	Virus.Win32.Agent
W32/Trojan.MUC	P2P-Worm.Win32.Agent.m	Win32.Worm.Amire.A	Trojan-Downloader.Win32.Small
W32/Trojan.MUD	P2P-Worm.Win32.Agent.m	Win32.Worm.Amire.B	P2P-Worm.Win32.Agent.m
W32/SillyWorm.PW	P2P-Worm.Win32.Agent.o	Win32.Worm.P2p.Agent.O	Nada encontrado
W32/Backdoor.KOH	Trojan.Win32.VB.alw	Trojan.Vb.ALW	Trojan.Win32.VB
W32/Heuristic-210!Eldorado	Nada encontrado	Gen:Trojan.Heur.PT.aiadaiHa9xpi	Trojan-Downloader.Win32.Small
W32/Heuristic-210!Eldorado	Email-Worm.Win32.generic	Generic.Malware.SPCTk.21E4927D	Win32.SuspectCrc
W32/SillyWorm.GS	Worm.Win32.VB.av	Win32.Worm.VB.AQ	Trojan-Downloader.Win32.Banload
W32/SillyWorm.MV	Worm.Win32.VB.bi	Win32.Worm.VB.BI	Worm.Win32.VB
W32/Heuristic-210!Eldorado	Trojan.Win32.Scar.atwl	Trojan.Generic.3674003	Trojan.Win32.Scar
W32/SillyWorm.PL	Hoax.Win32.BadJoke.VB.am	Trojan.Dropper.EX	Hoax.Win32.BadJoke.VB
W32/Trojan.CRQ	Trojan.Win32.Agent.un	Generic.Malware.dld!!.6490323D	Trojan.Win32.Agent
Malware!bdca	IRC-Worm.IRC.Pucker.a	IRC-Worm.Pucker.A	IRC-Worm.IRC.Pucker
W32/Scorvan.A	Worm.Win32.Scorvan	Generic.Malware.N!g.F190922B	Worm.Win32.Scorvan
W32/Heuristic-210!Eldorado	Nada encontrado	Trojan.Generic.3673998	Win32.SuspectCrc
W32/Trojan.BUI	Trojan.Win32.VB.afx	Trojan.Vb.AFX	Trojan.Win32.VB
W32/VB.OS	Virus.Win32.VB.z	Backdoor.981	Virus.Win32.VB
W32/Heuristic-210!Eldorado	Nada encontrado	Trojan.Generic.3674002	Win32.SuspectCrc
W32/SillyWorm.PL	Virus.Win32.VB.x	Nada encontrado	Virus.Win32.VB
W32/Trojan.BVI	Trojan.Win32.VB.acd	Trojan.VB.ACD	Trojan.Win32.VB
Nada encontrado	Virus.Win32.Zezal.a	Trojan.Generic.580332	Virus.Win32.Leon
Nada encontrado	Trojan.Win32.Genome.arxe	Gen:Win32.FileInfector.buX@aqPmRrN	Trojan.Win32.Genome
W32/TrojanSimulator.A	Nada encontrado	Application.TrojanSimulator	not-a-virus.Risktool.Trojansimulator
Nada encontrado	Nada encontrado	Nada encontrado	Nada encontrado

Kaspersky	NOD 32	PANDA	Quick Heal	SOPHOS
IRC-Worm.DOS.Funex.a	Funex.A worm	Nada encontrado	Nada encontrado	Nada encontrado
Trojan.Win32.VB.afw	Win32/VB.AFW	Nada encontrado	Trojan.VB.afw	Troj/VB-CKX
Trojan.DOS.DelSystem.l	DelSystem.L	Trj/DelSystem.P	Nada encontrado	Nada encontrado
Trojan.DOS.DelSystem.m	DelSystem.M	Trj/DelSystem.P	Nada encontrado	Nada encontrado
Virus.Win32.Agent.ab	unknown WIN32	Nada encontrado	W32.Agent.AB	W32/Ngvck-U
P2P-Worm.Win32.Agent.m	Win32/Agent.M worm	Nada encontrado	I-Worm.Agent.m	W32/Amire-A
P2P-Worm.Win32.Agent.m	unknown NewHeur_PE	Nada encontrado	I-Worm.Agent.m	W32/Amire-B
P2P-Worm.Win32.Agent.o	Win32/Agent.O worm	Nada encontrado	I-Worm.Agent.o	Mal/NafBot-A
Trojan.Win32.VB.alw	Win32/VB.ALW	Nada encontrado	Trojan.VB.alw	Troj/VB-BAF
TrojanDownloader.Win32.Small.s	unknown NewHeur_PE	Nada encontrado	Nada encontrado	Mal/Packer
Email-Worm.Win32.generic	unknown NewHeur_PE	Nada encontrado	Nada encontrado	Mal/TibsPk-A
Worm.Win32.VB.av	Win32/VB.NDA worm	Nada encontrado	Worm.VB.av	Mal/Behav-103
Worm.Win32.VB.bi	Win32/VB.NEG worm	Nada encontrado	Worm.VB.bi	W32/VB-CPH
Trojan.Win32.Scar.atwl	unknown NewHeur_PE	Nada encontrado	Trojan.Scar.atwl	Mal/TibsPk-A
Hoax.Win32.BadJoke.VB.am	Win32/VB.NFD worm	Nada encontrado	BadJoke.VB.am (Not a Virus)	Troj/VB-BBV
Trojan.Win32.Agent.un	Win32/Agent.UN	Trj/Downloader.MDW	Trojan.Agent.un	Troj/Agent-CVP
IRC-Worm.IRC.Pucker.a	Pucker.A worm	Generic	Nada encontrado	Nada encontrado
Worm.Win32.Scorvan	Win32/Scorvan.A worm	Nada encontrado	Worm.Scorvan	W32/Scorvan-A
Nada encontrado	unknown NewHeur_PE	Nada encontrado	Nada encontrado	Mal/Packer
Trojan.Win32.VB.afx	Win32/VB.AFX	Nada encontrado	Trojan.VB.afx	Troj/VB-QA
Virus.Win32.VB.z	Win32/VB.Z	Nada encontrado	Nada encontrado	W32/VB-BCF
Nada encontrado	unknown NewHeur_PE	Nada encontrado	Nada encontrado	Mal/Packer
Virus.Win32.VB.x	Win32/VB.NBC	Nada encontrado	Nada encontrado	W32/Pardis-B
Trojan.Win32.VB.acd	Win32/VB.ACD	Nada encontrado	Trojan.VB.acd	Troj/VB-BSZ
Virus.Win32.Zezal.a	Win32/Zezal	Nada encontrado	Nada encontrado	W32/Idas-A
Trojan.Win32.Genome.arxe	unknown CRYPT.WIN32	Nada encontrado	Nada encontrado	Mal/Generic-A
Nada encontrado	Win32/TrojanSimulator	Application/TrjSimulator	Trojan.Agent.ATV	Mal/Behav-053
Nada encontrado	Nada encontrado	Nada encontrado	Nada encontrado	Nada encontrado

VBA32	VirusBuster
VBA 32	VirusBuster
Nada encontrado	Nada encontrado
Trojan.Win32.VB.afw	Trojan.VB.EGH
Nada encontrado	Nada encontrado
Nada encontrado	Nada encontrado
Virus.Win32.Agent.ab	Nada encontrado
P2P-Worm.Win32.Agent.m	Packed/FSG
P2P-Worm.Win32.Agent.m	Packed/FSG
P2P-Worm.Win32.Agent.o	Packed/FSG
Trojan.Win32.VB.alw	Trojan.VB.VPE
MAS.Trojan.VB.01819	Packed/FSG
Nada encontrado	I-Worm.Iranvig.A
Worm.Win32.VB.av	Worm.VB.ACVQ
Worm.Win32.VB.bi	Worm.VB.DYF
Nada encontrado	Packed/FSG
Email-Worm.Win32.generic	Worm.VB.DWM
Trojan.Win32.Agent.un	Trojan.DL.Agent.CLW
Nada encontrado	Nada encontrado
Win32.Worm.Scorvan	Worm.Win32.Scorvan.A
Nada encontrado	Packed/FSG
MAS.Trojan.VB.01819	Trojan.VB.DWX
Virus.Win32.VB.Z	Nada encontrado
Nada encontrado	Packed/FSG
Virus.Win32.VB.x	Nada encontrado
Trojan.Win32.VB.acd	Trojan.VB.YEN
Nada encontrado	Nada encontrado
Unknown.Win32Virus	Worm.Agent.TKSZ
Nada encontrado	Nada encontrado
Nada encontrado	Nada encontrado

30	ise32.exe	Nada encontrado	Win32:Obfuscated-GDY	Downloader.Agent.AIQG	TR/Crypt.XPACK.Gen
31	bom76.exe	Trojan.Malware.Constructor	Nada encontrado	Constructor.UJ	KIT/Bom.76
32	vgyn6ewc.exe	Nada encontrado	Nada encontrado	Nada encontrado	TR/PSW.Frethog.128512.H.1
33	dmgr.exe	Nada encontrado	Win32:Rimecud-B	Worm/Generic.BACC	TR/Crypt.ZPACK.Gen
34	winmap32.exe	Nada encontrado	Win32:Rimecud-B	Worm/Generic.BACC	TR/Crypt.ZPACK.Gen
35	sys32.exe	Worm.Autorun.Dmh	Win32:AutoRun-AFL	Worm/Generic.HEQ	TR/Crypt.ULPM.Gen
36	psysnew.exe	Trojan.Vbkrypt.il	Win32:Malware-gen	Dropper.Generic.CJMM	TR/Dropper.Gen
37	bullmoose.exe	Trojan.Small.Cds	Nada encontrado	Generic15.BZXE	TR/Malex.6656F
38	He4HookInv.sys	Trojan.Rootkit.Agent.H	Win32:Trojan-gen	BackDoor.Agent.SL	RKIT/Rootkit.D.2
39	He4HookControl.exe	Trojan.Rootkit.Agent.H	Win32:Trojan-gen	Agent.FH	RKIT/Rootkit.D.1
40	injector.exe	Trojan.Fakegina.L	Win32:Trojan-gen	Generic.KLT	TR/FakeGina.L
41	sredir.zip	Riskware.Server-proxy.Small.A	Win32:Trojan-gen	Nada encontrado	TR/Crypt.XPACK.Gen
42	Unreal.exe	Trojan.Rootkit.Agent.Gv	Win32:Trojan-gen	BackDoor.Generic8.GAA	RKIT/Laernu.A.2

falso negativo		13	30	5	2
verdadeiro positivo		28	11	36	39

Packer.Pohernah.C	Trojan.Downloader-37161	Troj.Downloader.W32.Agent.obi	BackDoor.IRC.Flood.8
Trojan.Constructor.Bom.76	VirTool.W32.BoM.76	Nada encontrado	BOM.Generator.76
Nada encontrado	PUA.Packed.ASPack	Nada encontrado	Trojan.PWS.Wsgame.12661
Worm.P2P.Palevo.A	Worm.Palevo-640	Worm.Palevo-640	Win32.HLLW.Lime.5
Worm.P2P.Palevo.A	Worm.Palevo-640	Worm.Palevo-640	Win32.HLLW.Lime.5
Worm.Generic.60866	Trojan.Generic.Bredolab-2	IRC.W.W32.Small.u	Trojan.Inject.3265
Trojan.Generic.4335199	Nada encontrado	Nada encontrado	Trojan.Packed.19832
Trojan.Generic.2805155	Trojan.Agent-128610	Nada encontrado	Trojan.Siggen1.54357
Trojan.Rootkit.D	Trojan.Rootkit-136	Nada encontrado	Trojan.He4RootKit
Trojan.Rootkit.D	Trojan.Rootkit-135	Nada encontrado	Trojan.He4RootKit
Trojan.Generic.52006	Nada encontrado	Troj.W32.FakeGina.l	Tool.PassDump.7
Application.SckRedir.A	Nada encontrado	Server-Proxy.W32.Small.a	Tool.Proxy.2512
Rootkit.Unreal.A	Suspect.Trojan.Generic.FD-4	Nada encontrado	Trojan.NtRootKit.198

5	10	18	6
36	31	23	35

W32/Downldr2.BXDT	Worm.Win32.AutoRun.gmf	Packer.Pohernah.C	Packer.Pohernah.C
W32/VirTool	Constructor.Win32.Bom.76	Trojan.Constructor.Bom.76	Constructor.Win32.Bom.76
W32/Taterf.B!Generic	Nada encontrado	Nada encontrado	Worm.Win32.Taterf
W32/Palevo.A	P2P-Worm:W32/Palevo.M	Worm.P2P.Palevo.A	P2P-Worm.Win32.Palevo
W32/Palevo.A	P2P-Worm:W32/Palevo.M	Worm.P2P.Palevo.A	P2P-Worm.Win32.Palevo
W32/OnlineGames.AJ.gen!Eldorado	Worm.Win32.AutoRun.dmh	Worm.Generic.60866	Worm.Win32.AutoRun
W32/VBTrojan.Dropper.4!Maximus	Trojan.Win32.VBKrypt.il	Trojan.Generic.4335199	Trojan.Win32.Kreeper
Nada encontrado	Trojan.Win32.Small.cds	Trojan.Generic.2805155	Trojan.Win32.Small
W32/He4RootKit.A	Rootkit.Win32.Agent.h	Trojan.Rootkit.D	Rootkit.Win32.Agent
W32/He4RootKit.A	Rootkit.Win32.Agent.h	Trojan.Rootkit.D	Win32.SuspectService
W32/Fakegina.J	Trojan.Win32.FakeGina.l	Trojan.Generic.52006	Trojan.Win32.FakeGina
W32/Malware!eb70	not-a-virus:Server-Proxy.Win32.Small.a	Application.SckRedir.A	not-a-virus:Server-Proxy.Win32.Small
Nada encontrado	Rootkit.Win32.Agent.gv	Rootkit.Unreal.A	Trojan.IRC.Backdoor.SdBot4

5	6	4	2
36	35	37	39

Worm.Win32.AutoRun.gmf	Win32/AutoRun.KS worm	W32/Autorun.UT.worm	TrojanDownloader.Agent.obi	Mal/Krap-K
Constructor.Win32.Bom.76	Win32/Bom.76 Constructor	Constructor/Bom.A	Constructor.Bom.76 (Not a Virus)	Troj/Bom-76
Nada encontrado	Nada encontrado	Nada encontrado	Trojan.Agent.WD	Nada encontrado
P2P-Worm.Win32.Palevo.ann	Win32/Peerfrag.BG worm	Generic	Worm.Silly	W32/Autorun-AIC
P2P-Worm.Win32.Palevo.ann	Win32/Peerfrag.BG worm	Generic	Worm.Silly	W32/Autorun-AIC
Worm.Win32.AutoRun.dmh	Win32/AutoRun.LZ worm	Bck/Agent.IRW	Worm.AutoRun.dmh	Troj/Agent-GXK
Trojan.Win32.VBKrypt.il	Win32/Injector.BIM	W32/Agent.NNT.worm	Trojan.Meredrop	Mal/VBInject-D
Trojan.Win32.Small.cds	Win32/Agent.RCX	Nada encontrado	Trojan.Small.cds	Mal/Generic-A
Rootkit.Win32.Agent.h	Win32/Rootkit.Agent.H	Rootkit/He4.A	Rootkit.Agent.h	Troj/He4Hook-C
Rootkit.Win32.Agent.h	Win32/Rootkit.D	Rootkit/He4.A	Rootkit.Agent.h	Troj/He4Hook-C
Trojan.Win32.FakeGina.l	Win32/FakeGina.L	Nada encontrado	Trojan.FakeGina.l	Troj/FakeGin-B
not-a-virus:Server-Proxy.Win32.Small.a	Nada encontrado	Nada encontrado	Nada encontrado	Mal/Generic-A
Rootkit.Win32.Agent.gv	Win32/Rootkit	Generic	Rootkit.Agent.gv	Mal/Generic-A

5	3	27	14	6
36	38	14	27	35

Worm.Win32.AutoRun.aqla	Packed/Pohernah
Constructor.Win32.Bom.76	Constructor.Bom.F
MalwareScope.Worm.Viking.2	Nada encontrado
BScope.Backdoor.SdBot.ofx	Worm.Palevo.Gen!Pac
BScope.Backdoor.SdBot.ofx	Worm.Palevo.Gen!Pac
Virus.Win32.Agent.bc	Worm.Hamweg.C
MAS.Trojan.VB.01964	Trojan.Meredrop.WOF
Trojan.Win32.Small.cds	Trojan.Small.CUUV
Nada encontrado	Trojan.He4RootKit.A
Trojan.Win32.Rootkit.d	Rootkit.Agent.O
Trojan.Win32.FakeGina.02	Trojan.FakeGina.W
RiskWare.Proxy.Small.a	VirTool.SockRedir.A
Trojan.NtRootKit.198	Rootkit.Agent.DAZJ

12	11
29	30