INTRODUÇÃO A ANÁLISE FORENSE COMPUTACIONAL: DETECTANDO ROOTKITS EM AMBIENTE WINDOWS

Thiago Giroto Milani, Ricardo Slavov
Centro Universitário Hermínio Ometto - UNIARARAS, Araras – SP, Brasil
*e-mail: thiagogmilani@gmail.com

RESUMO

Atualmente existe tecnologia em praticamente qualquer tipo de atividade, com isso a segurança da informação torna-se mais necessária. Tudo em um ambiente digital é difícil de rastrear, necessitando de leis mais específicas e profissionais especializados em crime cibernético. Este artigo tem como objetivo o estudo da introdução a computação forense, mostrando a base para uma investigação digital e sua grande variedade de ataques computacionais, além de como detectar *rootkits*, em sistemas Windows, utilizando técnicas forenses e a base para a elaboração de um laudo pericial. Com isso mostrar a real dificuldade enfrentada por peritos para comprovar um crime em ambiente digital.

Palavras-chave
Segurança da Informação, Cyber-crime, Rootkit.

INTRODUÇÃO

A tecnologia vem crescendo significativamente desde o seu nascimento, e com isso não precisou de muito para que estivesse presente em praticamente todas as atividades no qual o ser humano desempenha, incluindo assim o âmbito judiciário, e criminal. Com esse crescimento e domínio da tecnologia diversas aplicações e funcionalidades surgem a todo o momento para facilitar e agilizar o dia-a-dia das pessoas, seja com o envio e recebimento de um *e-mail*, publicação de uma notícia em alguma revista eletrônica, *post* em redes sociais, efetuar uma compra, rastrear a localização de um objeto, ou até mesmo pagar uma conta por um portal de *Internet-Banking*.

Mas com toda essa evolução, praticidade e tecnologia sendo incorporada a quase todas as atividades e dispositivos, surge uma nova vertente de crimes, os crimes de informática, ou *Cyber*-Crimes. Esses *Cyber*-Criminosos atuam de diversas formas e com diversos objetivos diferentes no mundo digital. O principal crime de informática, ou o mais comum nos dias de hoje é o roubo de informação, sendo ela de diversos tipos: desde fotos pessoais ou íntimas para uma futura chantagem ou extorsão, até mesmo senhas bancárias e de cartões de credito.

Um *cyber*-criminoso, para efetuar esses crimes de informática utiliza de diversas técnicas e ferramentas, alguns até mesmo criam suas próprias ferramentas para invasão, e roubo de informação. *Rootkit* é uma dessas ferramentas onde facilita a invasão do equipamento computacional para que possa ser cometido o crime, e é exatamente em *rootkits* que este artigo irá focar; iremos tratar de *rootkits apenas* em ambiente Windows.

> *"A perícia forense é o suporte técnico ao judiciário, realizado preferencialmente por pessoas com formação e capacidade para isso nas mais diversas áreas do conhecimento..."* (GALVÃO R. K. M., 2013).

Da mesma forma acontece com a área da perícia forense computacional. Profissionais formados em diversas áreas da tecnologia da informação se especializam e se capacitam para

efetuar a coleta e a análise de dados digitais, e auxiliar o poder judiciário.

REVISÃO BIBLIOGRÁFICA

1. Um pouco de história.

O primeiro computador foi desenvolvido com a finalidade de executar cálculos balísticos a pedido das forças armadas norte-americanas, durante a segunda guerra mundial. O projeto foi iniciado em 1943, porém só ficou pronto um ano após o fim da guerra (1946). Ele recebeu o nome de ENIAC – *Eletronic Numerical Integrator and Computer*, composto por aproximadamente 17.000 válvulas termiônicas e capacidade de processar 5.000 operações por segundo. Alguns anos depois em 1965 um pesquisador e co-fundador da Intel, Gordon Moore "...observou que o número de transistores que podiam ser impressos em uma pastilha (chip), dobrava a cada ano..." (EUHRARA, M., TSAI, D., 2011), algum tempo depois ele percebeu que o número de transistores dobrava a cada dois anos, e não a cada um ano. Com isso a evolução dos processadores e a capacidade de processamento deles cresceu rapidamente até os dias de hoje onde temos processadores com capacidade de processar mais de 5 milhões de operações por segundo.

Após a popularização da internet, e os computadores ficando menores e chegando cada vez mais dentro de casa, começou a surgir várias empresas desenvolvendo softwares e vendendo serviços de informática. Em meados de 1990, várias empresas e principalmente o governo norte americano vinha migrando todas as atividades possíveis para o mundo digital. Trazendo assim muito conteúdo sigiloso e pessoal para "a era digital", onde inicialmente aparentava ser mais seguro e confiável, do que os antigos métodos de arquivos em papel e tinta.

2. O que é Computação Forense?

> *"A Forense Computacional tem como objetivo, a partir de métodos científicos e sistemáticos, reconstruir as ações executadas nos diversos ativos de tecnologia utilizados em cyber-crimes."* (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; et. All. 2008, pag. 4).
>
> *"Um dos primeiros casos de descoberta de fraudes a partir de experimentos científicos é relatado pelo historiador romano Vitrúvio, segundo o qual Arquimedes foi chamado pelo rei Hieron para atestar que a coroa encomendada junto a um artesão local não era composta pela quantidade de ouro combinada previamente entre as partes. Embora existisse essa suspeita, o rei Hieron não tinha evidências que lhe permitissem acusar o fraudador e, portanto, atribuiu a tarefa de investigação sobre o caso a Arquimedes que, depois de algum tempo e quase que por acaso, formulou a teoria do peso específico dos corpos."* (INMAN, K., RUDIN, N., 2000).

Segundo (GALVÃO R. K. M., 2013) o termo perícia forense é muito utilizada na área criminal, quando se é necessária uma análise mais detalhada sobre um determinado crime, podendo ser de vários tipos, desde análise forense de um homicídio, roubo, ou crime que envolva equipamentos computacionais.

Conforme o CPP (Código de Processo Penal Brasileiro) em sua oitava edição, no artigo 158 que diz:

> "Quando a informação deixar vestígios será indispensável o exame de corpo de delito, direto ou indireto, não podendo supri-lo a confissão do acusado."

Com isso surge a necessidade de um perito que investigue e crie um laudo detalhado de interesse da justiça, conforme os caputs dos artigos 159 e 160 do CPP que respectivamente dizem: "O exame de corpo de delito e outras perícias serão realizadas por perito oficial, portador de diploma de ensino superior." e "Os peritos elaborarão o laudo pericial, no qual descreverão minunciosamente o que examinaram e responderão aos quesitos formulados."

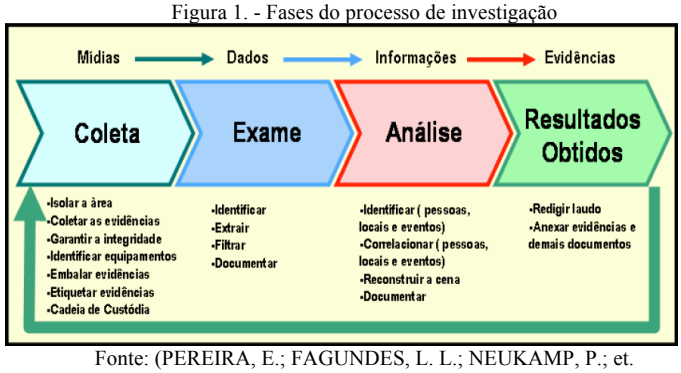
(Exemplo do formulário de cadeia de custódia está no ANEXO I).

> *"Portanto, a Computação Forense tem como objetivo principal determinar à dinâmica, a materialidade e a autoria de ilícitos ligados à área de informática, tendo como questão principal a identificação e o processamento de evidências digitais..."(* ELEUTÉRIO, P. M. S.; MACHADO, M. P., 2011).

De acordo com (KENT. Et. All., 2006) e (KRUSE, 2001) as fases de um processo de investigação forense são :

A. **Coleta de dados:** nessa fase os dados relacionados a uma evidência devem ser coletados e a integridade do mesmo deve ser preservada, posteriormente, os equipamentos devem ser identificados, devidamente embalados, etiquetados e suas identificações registradas;
B. **Exame de dados:** nessa segunda fase são selecionadas e utilizadas ferramentas e técnicas apropriadas a cada tipo de dado coletado, a fim de identificar e extrair as informações relevantes ao caso que está sendo investigado, mais sempre com a preocupação de manter a integridade dos dados;
C. **Análise das informações:** a terceira etapa se refere à análise dos dados filtrados na etapa anterior, cujo objetivo é obter informações relevantes e úteis que possam responder às perguntas que deram origem à investigação;
D. **Interpretação dos resultados:** na ultima fase do processo de investigação é gerado um relatório no qual deve estar descrito os procedimentos realizados e seus respectivos resultados obtidos.

Figura 1. - Fases do processo de investigação



Fonte: (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; et. All. 2008).

Devido à informática ser uma área de estudo muito ampla e complexa, existem diversos meios e mecanismos de análise, e coleta de evidências digitais, segundo o autor (GALVÃO, R. K. M., 2013) existem quatro principais terminologias da análises e coleta de evidências digitais:

- **Mídia de prova:** engloba todos os objetos, dispositivos e mídias alvos da investigação;
- **Mídia de destino:** imagem pericial fidedigna das mídias de provas armazenadas com proteção contra alteração;
- **Análise ao vivo:** análise pericial realizada diretamente sobre as mídias de provas (geralmente acontece quando não se dispõe de recursos, e/ou tempo para a adequada geração de mídia de destino);
- **Análise post-mortem (*offline*):** metodologia de perícia mais utilizada e recomendada onde à análise é feita sobre a mídia de prova ou sobre uma cópia, permitindo maior flexibilidade nos procedimentos adotados para a análise dos dados.

3. Softwares Maliciosos (*Malwares*), Vírus e Ataques Computacionais.

"Quando se fala em *Software* Malicioso (*malware*), a primeira palavra que vem a cabeça é vírus." (VECCHIA, E. D., 2014). Mas vírus é apenas um das categorias conhecidas, além de *malware*, vírus, *spyware*, cavalo de troia (*trojan horse)*, e tantos outros existem várias técnicas de invasão em sistemas e redes, além da engenharia

social, onde *hackers* e *crackers* as utilizam em diversas maneiras e combinações diferentes para obterem acessos ilegítimos a sistemas e redes "...realizando a coleta de informações sem a devida autorização." (VECCHIA, E. D., 2014). Segundo (SKOUDIS, E.; ZELTSER, L.;2003) "código malicioso são conjuntos de instruções em um computador e que fazem o sistema realizar algo que um atacante deseja." Com os tempos de hoje essa afirmação se torna um pouco vaga, visto a quantidade de derivações que os *malwares* têm.

Conforme (KENT. Et. All., 2006) e (KRUSE, 2001), além dos *softwares* maliciosos (*malwares*) os *hackers* e *crackers*, constantemente estão tentando criar novas aplicações, vírus, ou até mesmo metodologias para invadir e explorar vulnerabilidades dos sistemas operacionais, porém é mais comum ataques e vulnerabilidades exploradas em ambiente Windows, devido a ser um dos sistemas operacionais mais utilizados.

3.1. Vírus

> *"O primeiro código com capacidade de se auto replicar de que se tem notícia surgiu em 1962, nos laboratórios Bell, com o jogo Darwin, onde programas "lutavam" entre si para sobreviver."* (NULL A, 1971).

Nos dias de hoje vírus "é a categoria mais conhecida de *malwares*, tanto que a maioria dos *softwares* que detectam seus diversos tipos são chamados de antivírus." (VECCHIA, E. D., 2014). O que muitos ainda se enganam é que um vírus não necessariamente é um arquivo executável (.exe), ele pode ser escondido dentro de praticamente qualquer tipo de arquivo, podem estar em arquivos do tipo pdf, jpg ou "...embutidos em documentos de texto e planilhas eletrônicas." (VECCHIA, E. D., 2014).

Quando um vírus é executado ele pode reagir de três formas possíveis: "(a) executar no programa hospedeiro e propagar a infecção, (b) danificar o sistema ou (c) imitar o programa hospedeiro." (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; et. All. 2008, pag. 6).

Para evitar esse tipo de *software* malicioso, recomenda-se que antes de abrir um arquivou ou anexo de um e-mail, execute um *software* de antivírus. Apenas ter o antivírus instalado não é sinônimo de cem por cento de proteção, deve-se manter sua "... base de assinaturas de vacinas constantemente atualizadas..." (VECCHIA, E. D., 2014) para que possa se manter o mais protegido possível, pois novos *malwares* são criados todos os dias.

3.2. *Spyware* e *Adware*

Como o próprio nome diz, *spyware* é um software malicioso para espionar um dispositivo computacional, ele pode ser usado de maneira legal em empresas com a ciência de todos os funcionários, ou de maneira ilegal, quando instalado em um computador sem que seu dono saiba, com o intuito de espionar seu uso. "O software pode ser configurado para analisar apenas algumas atividades..." (VECCHIA, E. D., 2014) e seu conteúdo enviado para um administrador ou salvo em algum local seguro em sua mídia.

Levando em conta todas as categorias de *malwares*, o *spyware* é o mais comum de ser encontrado segundo o autor (WEISS, A., 2005), uma pesquisa conduzida pela Dell em setembro de 2004 estimou que aproximadamente 90% dos PCs com o sistema operacional Windows possuíam no mínimo um *spyware*. "Este tipo de *software* foi responsável por metade das falhas em ambiente Windows reportado por usuários da Microsoft" (SHUKLA, S.; NAH, F. F., 2005). "Outro estudo aponta uma média de 25 *spywares* por computador." (SIPIER, J. C.; WARD, B. T.; ROSELLI, G. R., 2005).

3.3. KeyLogger

"*Keylogger* é um tipo de *spyware*, mas, pelo fato de ser amplamente utilizado, pode ser considerada uma nova categoria. (VECCHIA, E. D., 2014). "*Keylogger* são definidos como um tipo de *spyware* cujo a finalidade é capturar tudo que for digitado em um computador." (CERT.br, 2007).

Hoje em dia todos os usuários de computador utilizam ao menos um sistema que exija usuário e

senha, ou até mesmo que utilize um protocolo de criptografia na internet (HTTPS). "Porém, pode haver um *software* instalado no dispositivo que está em uso, capaz de capturar tudo o que é digitado (teclado) ou clicado (mouse), um *keylogger*." (VECCHIA, E. D., 2014). Sendo assim nenhum mecanismo de segurança para senhas é totalmente seguro.

De acordo com (SECURITYFOCUS, 2007), os *keyloggers* podem ser classificados em três diferentes tipos:

A. *Hardware keylogger*: trata-se de um dispositivo físico posicionado entre o teclado e o computador da vítima. Apesar de serem rápidos (por possuírem um *hardware* dedicado para executar sua função) e por não serem detectados por mecanismos como anti-vírus e anti-*spyware*, os mesmos podem ser detectados visualmente por uma pessoa. Além disso, possuem espaços de armazenamento limitados e necessitam de acesso físico a máquina (vítima) para serem instalados;

Figura 2. - *Hardware Keyloger*

Fonte: (https://www.keelog.com/pt/hardware_keylogger.html)

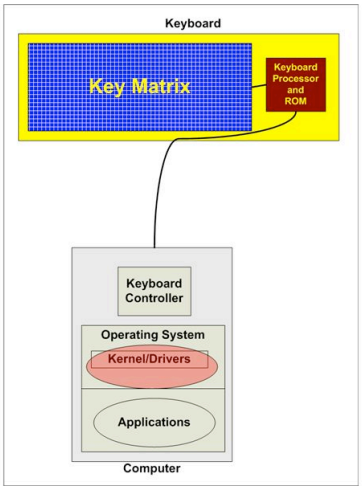
B. *Sofware keylogger* usando um mecanismo de *hooking*: um *hook* trata-se de uma rotina que tem como objetivo "ficar no meio do caminho" do tratamento normal da execução de informações do Sistema Operacional (SO). Para isso, os programadores utilizam funções disponibilizadas pela API (*Aplication Program Interface*) do SO. Essas funções são responsáveis por capturar as mensagens do sistema (assim como as teclas que são pressionadas) antes que as mesmas sejam tratadas pelas devidas rotinas de tratamento. *Keyloggers* desse tipo normalmente possuem um módulo executável, que dispara a execução do aplicativo, e uma biblioteca que contém as rotinas para a captura das informações desejadas. Esses *keyloggers* podem ser instalados remotamente, no entanto, são os mais lentos e facilmente detectáveis por programas como anti-vírus e anti-*spywares*;

Figura 3. - *Software Keylogger*

Fonte: Próprio Autor

C. *Kernel keylogger*: este tipo de *keylogger* trabalha no nível do *kernel* e usa suas próprias rotinas para receber os dados diretamente dos dispositivos de entrada (no caso, o teclado). É o método mais difícil de ser desenvolvido (por exigir um elevado conhecimento de programação) e também de ser detectado (por substituir as rotinas padrão do SO e serem inicializados como parte do próprio sistema). Pelo fato de trabalharem no núcleo do sistema, não são capazes de capturar informações que são trocadas diretamente no nível de aplicações (exemplo: operações de copiar e colar e operação de auto completar).

Figura 4. - Kernel Keylogger



Fonte: (http://www.brighthub.com/computing/smb-security/articles/68608.aspx)

Segundo o autor (VECCHIA, E. D., 2014) muitos softwares anti-*malware* ou anti-*spyware* conseguem detectar os *keylogger*, e bloqueá-los para que não consigam capturar o conteúdo digitado ou clicado, por isso é de grande importância sempre manter seus sistemas de segurança devidamente atualizados.

3.4. Backdoor

Um *backdoor,* ou porta dos fundos, é um dos mecanismos mais utilizados por *hackers* e *crackers* para voltar à maquina invadida sempre que quiser. Segundo (SKOUDIS, E.; ZELTSER, L.;2003) *backdoor* é um *software* que permite uma "entrada pelos fundos".

"Muitos *hackers* e *crackers* invadem dispositivos depois de várias tentativas e não querem ter o mesmo trabalho para retornar a ter o acesso." (VECCHIA, E. D., 2014), e para isso, instalam softwares que permitem acesso sem a autorização do usuário através de portas pouco comum e difícil de se detectar, "...sempre que o dispositivo for inicializado, uma mensagem seja enviada a ele, comunicando que o endereço IP está sendo usado." (VECCHIA, E. D., 2014), dessa forma o *hacker* ou cracker saberá o endereço IP da máquina na qual instalou o *backdoor*, e poderá acessa-la novamente.

Segundo (ZHANG, Y; PAXSON, V., 2000) normalmente um *backdoor* opera sobre o protocolo *Telnet*, *Rlogin* ou *SSH* e tipicamente fornece uma das seguintes funcionalidades ao atacante:

A. **Aumento dos Privilégios Locais (*Local Escalation of Privileges*):** permite que um usuário normal execute programas com privilégios de super-usuário;

B. **Execução Remota de Comandos:** permite que o atacante envie comandos para a máquina alvo e obtenha as respostas geradas pela execução dos mesmos;

C. **Acesso Remoto à Linha de Comando:** permite que o atacante utilize um *shell* remoto na máquina alvo, de onde poderá realizar qualquer operação como se estivesse utilizando o teclado em frente à máquina real.

D. **Controle Remoto da Interface Gráfica:** permite ao atacante observar e interferir na interface gráfica à qual o usuário local está conectado, fornecendo assim acesso pleno a maquina.

"Outro exemplo de programas que podem ser utilizados como *backdoor* é o *Virtual Network Computing*" (VNC). (CAMBRIDGE, A. L., 2007) pois ele pode proporcionar acesso remoto total ao dispositivo no qual foi instalado, dando assim permissão total ao atacante no dispositivo alvo.

3.5. Worms

Um *worm* é muito parecido com um vírus, porém com finalidades diferentes, "*Worm* é um *software* capaz de se propagar automaticamente através de redes, enviando cópias de si mesmo." (VECCHIA, E. D., 2014). Porém ele não é introduzido em outros *softwares*, e não precisa ser executado para se propagar pela rede, "Sua propagação dá-se através da exploração de vulnerabilidades..." (VECCHIA, E. D., 2014), a finalidade dele é consumir grande parte dos recursos do computador infectado e da rede, causando uma queda no desempenho.

> *"Atualmente, os worms são classificados como uma das maiores, ameaças virtuais, chegando a atingir 65% dos incidentes de segurança reportados ao Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil (CERT.br) no período de Janeiro a Março de 2007."* (CERT.br, 2007).

3.6. Bot e Bootnet

"O termo *bot* originou-se de *robot* (robô). Isso porque um robô é criado para obedecer, e esta categoria de *malware* faz a mesma coisa..." (VECCHIA, E. D., 2014). Ou seja, um computador infectado se transforma em um zumbi (*bot*) e executa comandos enviados por um mestre. Segundo (HOLZ, T., 2005) há três atributos que caracterizam um *bot* (nome derivado de (*Robot*)): a existência de um controle remoto, a implementação de vários comandos e um

mecanismo de espalhamento, que permite ao *bot* espalhar-se ainda mais.

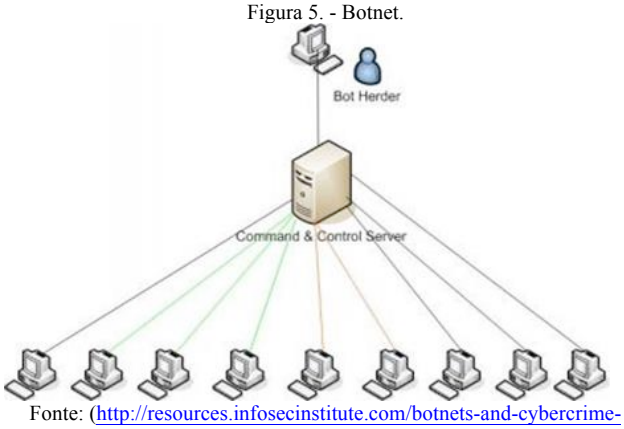
> *"A família de bots mais conhecida é provavelmente a família Agobot (também conhecida como Gaobot). Seu código é escrito em C/C++, com suporte a multe plataforma."* (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; et. All. 2008, pag. 9).

Segundo (SOPHOS, 2007) existem mais de mil variantes do *Agobot* conhecidas. Quando iniciado o *bot* tenta se conectar a um endereço previamente cadastrado e realizar um teste de velocidade, assim sendo mais fácil contabilizar quantos *bots* existem.

Em (MCLAUGHLIN, L., 2004), os autores apontam um estudo que estimava, em 2004, o *Phatbot* possuía uma rede de aproximadamente 400.000 *bots*. Normalmente um *bot*, se conecta a uma rede IRC (*Internet Relay Chat*) e fica aguardando instruções em um canal específico, quando enviado o comando por seu mestre ele executa a ação solicitada.

Essas *botnet* ou *bot-network,* como são chamadas as redes de *bot*, são muito utilizadas para ataques de negação de serviço (DoS), envio *de e-mail* de *phishing* (ver 3.8), *spam* ou para mascarar outros tipos de ataques e dificultar o rastreamento do verdadeiro atacante.

Figura 5. - Botnet.



Fonte: (http://resources.infosecinstitute.com/botnets-and-cybercrime-introduction/).

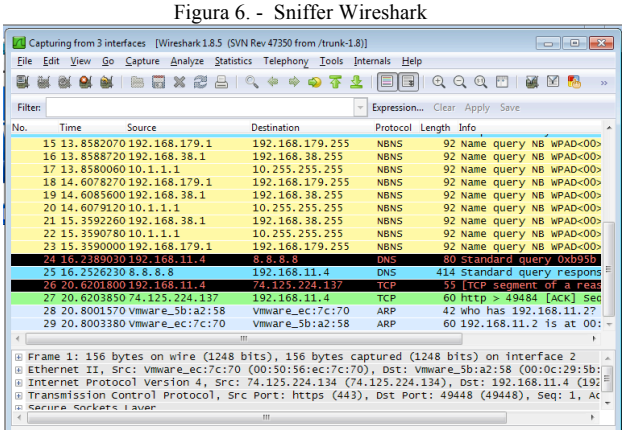
3.7. Sniffer

"Um *sniffer* tem como finalidade capturar pacotes que chegam até uma interface de rede de um dispositivo." (VECCHIA, E. D., 2014). As interfaces de rede são pré-programadas para descartar os pacotes que não são endereçados a elas. O *sniffer* tem a capacidade de alterar o estado da interface de rede para promíscuo, ou seja, passam a aceitar todos os pacotes da rede, até mesmo os que não estão endereçados a ele.

Com os pacotes capturados o atacante pode analisa-los detalhadamente e conseguir extrair dados sigilosos do usuário. "Se dados sensíveis, como dados de cartões de crédito, dados bancários, *e-mails*, senhas, forem trafegados sem nenhum tipo de proteção, esses serão capturados e visualizados sem nenhum problema pelo *sniffer*" (VECCHIA, E. D., 2014). Um software *sniffer* muito utilizado e conhecido é o *Wireshark*.

Porém para um atacante utilizar um *sniffer* de maneira eficiente, é preciso saber exatamente onde coloca-lo, pois ele apenas escuta o que está passando pela placa de rede do dispositivo em que está instalado.

Figura 6. - Sniffer Wireshark



Fonte: Próprio Autor

3.8. Engenharia Social e *Phishing*

A engenharia social está cada vez mais frequente, e trata-se da "arte de enganar". "São técnicas de persuasão utilizadas para convencer a pessoa a fornecer informações, geralmente sensíveis, como dados bancários, senhas, endereços, entre outros." (VECCHIA, E. D., 2014). Antigamente esse método de golpe era utilizando o telefone, porém com a era digital é muito mais fácil e prático enviar milhares de mensagens eletrônicas, para vários destinatários em pouco tempo.

> *"Muitos criminosos apelam para técnicas de engenharia social antes de tentativas de invasão de sistemas, pois é muito mais fácil enganar pessoas do que possuir conhecimento avançado..."* (VECCHIA, E. D., 2014).

Também são conhecidos como *phishing* os golpes onde envolvem a tentativa de roubos financeiros. "Por exemplo, um e-mail pode ser enviado com um formulário a várias pessoas, alegando ser de um banco e que é necessário uma atualização dos dados, entre eles a senha." (VECCHIA, E. D., 2014). Uma vez preenchido o formulário e enviado, o criminoso pode utilizar esses dados como quiser, para fazer movimentações bancarias.

3.9. Pharming

> *"Pharming é o termo atribuído ao ataque que compromete o serviço de DNS, fazendo com que um endereço (URL) seja traduzido para um endereço IP incorreto."* (VECCHIA, E. D., 2014).

Esse ataque pode ser feito em um servidor de DNS, como em uma máquina específica. "No caso da máquina da vítima, um *malware* pode, por exemplo, editar o arquivo que possui configurações de tradução nome/endereço IP." (VECCHIA, E. D., 2014). E assim quando a vítima tentar acessar um site de banco será redirecionado para um endereço IP onde poderá conter um site idêntico ao verdadeiro, porém preparado para capturar senhas e dados sensíveis.

4. O que é *Rootkit* (para que servem?)

Um *rootkit*, como o próprio nome diz, é um *kit* para o usuário *root* (administrador), no qual "o *hacker* ou *cracker* experiente procura utilizar mecanismos para destruir seus rastros e assegurar sua presença no dispositivo computacional" (VECCHIA, E. D., 2014) e para que consiga fazer isso, utiliza de um *rootkit*, no qual tem sua finalidade esconder *softwares* dentro do sistema para que não seja descoberto, e facilite para que o invasor volte a invadir a maquina sempre que quiser.

Rootkit pode ser definido como "um programa ou um conjunto de programas usado por um atacante para que o mesmo consiga ocultar sua presença em um determinado sistema e, ainda, para permitir acesso futuro a esse sistema." (KLAUS , S., NELSON, M., 2001).

Perceba que os *rootkits* não podem dar acesso de super administrador (*root*) ao atacante, (como no caso de um *exploit*) ele apenas mantém o privilégio ocultos em seus acessos futuros. (MICROSOFT[a], 2007). Segundo (KLAUS , S., NELSON, M., 2001), os *rootkits* podem ser classificados em dois tipos: os *rootkits* tradicionais, e os baseados em LKM (*Loadable Kernal Modules*).

4.1. Rootkits Tradicionais

> *"Os rootkits tradicionais começaram a ser desenvolvidos em meados de 1994 e são caracterizados por comandos modificados do sistema, como ls, ps, ifconfig, netstat,"* (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; Et. All., 2008, pag.10) entre outros.

Esses comandos começaram a ser programados e alterados com a finalidade de esconder dos administradores, dos sistemas, arquivos e processos. "No caso do *ifconfig*, por exemplo, o programa original é modificado e substituído por uma versão que oculta o fato de uma determinada interface de rede estar sendo executada em modo promíscuo." (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; Et. All., 2008, pag. 10).

Esses *rootkits* são mais facilmente detectados nos sistemas, "para isso é preciso fazer uso de programas específicos," (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; Et. All., 2008, pag. 10) no qual irão executar uma varredura no sistema original e qualquer alteração

irá ser mostrada ao perito que estiver executando a análise.

4.2. Rootkits LKM

"Já os *rootkits* LKM começaram a ser publicados em meados de 1997" (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; Et. All., 2008, pag. 15). Esse código malicioso funciona juntamente com o *kernel* do sistema, ou seja, uma vez infectada a maquina o código malicioso estará sendo executando juntamente com a inicialização do sistema.

> *"O processo de detecção desses malwares é muito mais difícil se comparado ao processo de detecção de rootkits tradicionais, pois os comandos do sistema continuam inalterados e o próprio kernel responderá às requisições."* (PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; Et. All., 2008, pag. 15).

5. Processos e *Threads* do Windows

Todo sistema operacional se organiza através de processos, para que possam ser executados e executar os demais programas. Independente do sistema operacional, são necessários processos para que haja essa organização no momento da execução.

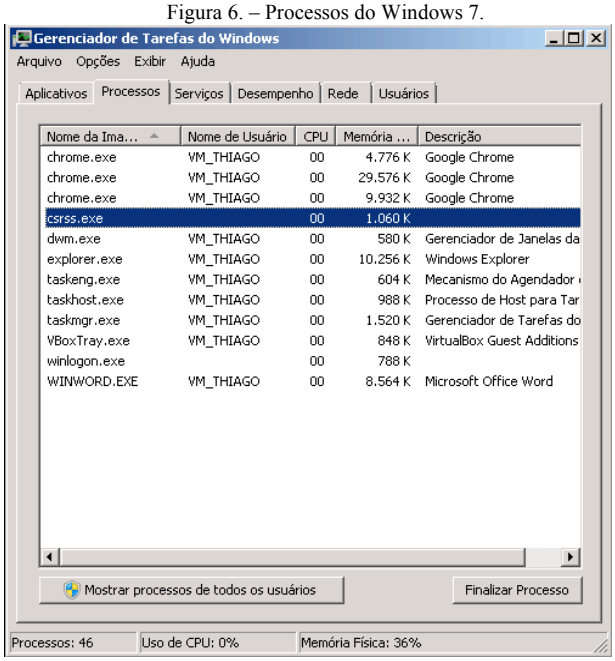
"Processo é diferente de programa, visto que um programa pode ser repetidamente executado gerando vários processos." (BARCELAR R. R., 2013).

> *"Cada processo fornece os recursos necessários para executar um programa. Um processo tem um espaço virtual de endereço, código executável, abertura de identificadores para objetos do sistema, um contexto de segurança, um identificador único de processo, variáveis de ambiente, uma classe de prioridade, tamanho mínimo e máximo do conjunto de trabalho, e pelo menos uma thread de execução. Cada processo é iniciado com uma única thread, muitas vezes chamado o primary thread, mas pode criar threads adicionais a partir de qualquer um de suas threads. Uma aplicação consiste de uma ou mais processos. Um processo, em termos mais simples, é um programa em execução. Uma ou mais threads são executado no contexto do processo. Uma thread é a unidade básica para a qual o sistema operacional aloca tempo do processador. Uma thread pode executar qualquer parte do código do processo, incluindo pedações que estão sendo executadas atualmente por outra thread."* (MICROSOFT, 2016).

Processo é um programa em execução, e possui os seguintes componentes: sessão de texto (código), contador de instruções, pilha e sessão de dados. Além disso, um processo pode conter várias linhas de controle (*threads*), onde "permite, por exemplo, que um editor de texto realizar uma verificação ortográfica ao mesmo tempo em que o usuário digita caracteres" (BARCELAR R. R., 2013). *Hackers* e *crackers* fazem um grande uso de *threads* pois é muito mais complicado ver, ou analisar, seu conteúdo ligado a vários processos ao mesmo tempo, e ficar escondido dos métodos tradicionais de visualização dos processos.

Todo processo tem seu ID, ou seja, seu código de identificação, código utilizado pelo sistema operacional para identificar o processo e seu conteúdo a ser executado, porém não é visível ao usuário como mostra a figura 6.

Figura 6. – Processos do Windows 7.



Fonte: Próprio Autor

Ainda sobre a Figura 6 podemos visualizar que o usuário comum tem acesso apenas ao nome do programa em execução, nomes dos usuários do S.O. à quem pertence o processo, quantidade de uso da CPU, quantidade de memória RAM utilizada e a descrição do processo, no qual normalmente faz referencia a empresa proprietária do programa a quem o processo em execução pertence. Assim que o programa finaliza sua execução, ou o mesmo é fechado pelo usuário os processos relacionados a ele são finalizados também.

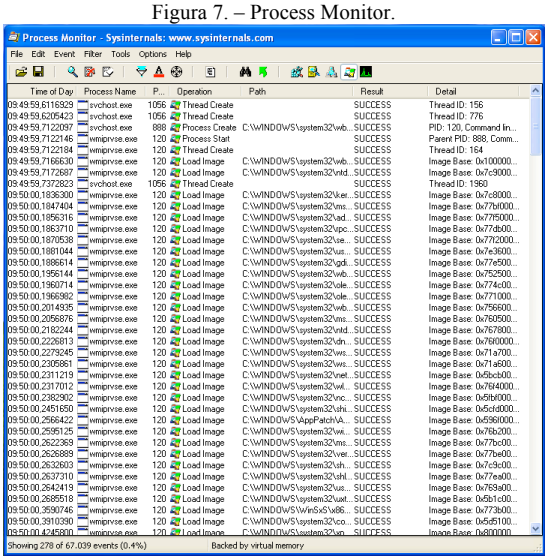
Um mecanismo muito utilizado em plataformas Windows quando uma determinada aplicação para de responder é forçar a finalização dos processos relacionados a aplicação, encerrando assim todos os processos e *threads* relacionados aquela aplicação.

6. Processos Maliciosos

Todo programa, seja malicioso ou não gera um ou mais processos no sistema operacional, independente se está na "cara do usuário" ou não, dessa forma todo *software* malicioso, seja um vírus, *worm*, *keylogger* ou qualquer outro programa malicioso pode ser descoberto através dos processos em execução como mostra a figura 6. Já foi explicado no tópico anterior, o gerenciador de tarefas do próprio Windows não é muito eficiente para mostrar os processos e *threads* em execução, facilitando então para um *hacker* ou *cracker* alterarem os dados básicos da execução do seu *software* malicioso.

Existem várias ferramentas específicas para verificação de processos, onde possibilita a visualização de todos os processos e *threads* em execução no sistema operacional, como mostra a figura 7.

Figura 7. – Process Monitor.



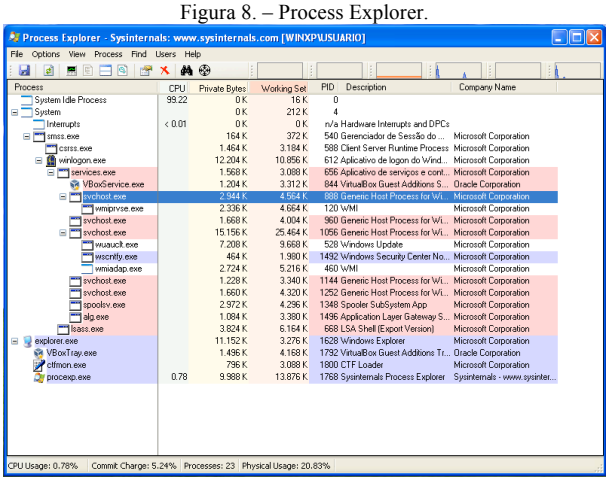
Fonte: Próprio Autor

O *software* em demonstração na figura 7 é o "*Process Monitor*", desenvolvido para o monitoramento detalhado dos processos e *threads* em execução, ele é composto de várias funcionalidades para filtrar alguns processos e *threads* específicas do sistema ou de terceiros, além de processos que estão utilizando conexão com a rede. Dessa forma é mais fácil verificar e encontrar um processo malicioso, no qual esteja usando um recurso específico do sistema operacional, como um *sniffer* de rede que estará utilizando sua conexão de rede, desse modo filtrar pelos processos em rede e executar uma varredura para encontrar o suposto *software* malicioso.

Também pode ser utilizado o *software "Process Explorer"* como mostra a figura 8, também desenvolvido para a análise mais profunda de processos em execução no Windows.

Figura 8. – Process Explorer.

Figura 8. – Process Explorer.



Fonte: Próprio Autor

Diferente do *Process Monitor*, ele identifica os processos e os demarca com cores para facilitar a visualização, e permite que o perito além de analisar os processos, execute ações em cima do processo, como finalizar ou pausar a execução, dando mais detalhes e ferramentas para o perito na hora de uma análise. Em determinados casos são necessárias combinações de diversas ferramentas até que se consiga detectar e analisar algo malicioso no sistema. Porém para que seja eficiente a varredura de processos maliciosos são necessários conhecimento dos processos básicos do sistema operacional em análise.

Com essas e outras ferramentas um perito forense consegue verificar e analisar um sistema e determinar se existe ou não softwares maliciosos sendo executados; a partir disso, tomar as medidas cabíveis para o laudo, ou solução.

7. Como detectar *Rootkits* em Windows

"Desde a década de 80 foram desenvolvidos programas para ocultar a presença de atacantes em um sistema computacional e também permitir seu acesso futuro a esse sistema." (PEREIRA, 2008, pag. 15).

A detecção de *rootkits* requer ferramentas específicas, que possa verificar o sistema operacional e detectar *softwares* maliciosos sendo executados por trás de processos e *threads*. Essas ferramentas funcionam basicamente executando uma varredura para detectar processos ou *threads* modificadas ou ocultas em processos no sistema operacional.

Foram testadas as três ferramentas mais conhecidas para a detecção de *rootkit* em ambiente Windows, onde ambas as três são específicas para essa finalidade.

7.1. Principais ferramentas para detecção de *rootkit* em Windows

Existem várias ferramentas para detecção de *malwares*, vírus e outros softwares maliciosos. Com os *rootkits* não é diferente, existem vários *softwares* para detectar *rootkits*. Todos se baseiam em varredura de processos em execução, análise de sua origem e até mesmo quais componentes do sistema operacional está afetando.

Com essa varredura é possível identificar quais processos ou *threads* foram alteradas para esconder um *malware,* ou, liberar portas de acesso ocultas para *hackers* e *crackers* acessarem quando quiserem. Foram estudadas as três ferramentas mais conhecidas para detecção de *rootkits*, são elas: *AVG gmer v2.2.19882, MCAfee Rootkit Remover v0.8.9.209, e o Kaskersy TDSKill v3.1.0.9*.

Todos os três *softwares* estudados são compatíveis com o sistema operacional da Microsoft Corporation desde sua versão Windows XP x86 até a versão Windows 10 x64.

Para maior segurança e resultados mais confiáveis foram testados em ambiente preparado e controlado, com imagens do Windows 10 x64 instaladas no *VirtualBox* v5.0.26 para *macOS Sierra 10.12*.

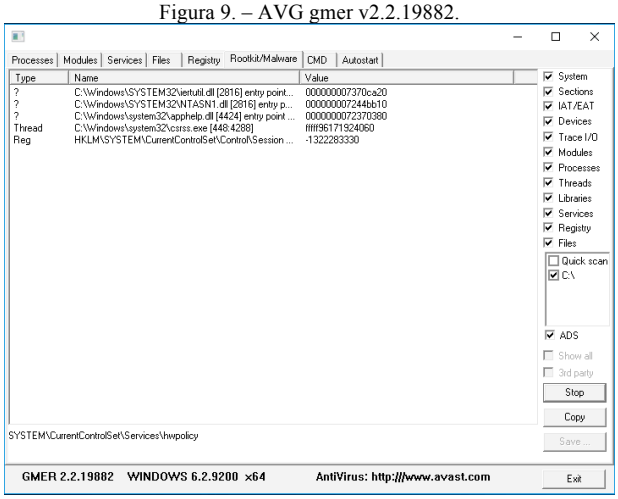
O primeiro *software* estudado foi o *AVGgmer*, composto de uma interface intuitiva e de fácil usabilidade. Logo quando executado como administrador, já inicia automaticamente um *Quick scan,* varredura rápida no sistema a fim de encontrar algum *malware*.

O *AVGgmer* é composto por várias ferramentas, desde visualização de processos em execução, até mesmo um *prompt* exclusivo, caso seja necessário comandos para reparos.

Quando o *scanner* de *rootkits* do *AVGgmer* é finalizado ele retorna e exibe um

relatório no qual pode ser exportado contendo os detalhes da varredura. (ANEXO II)

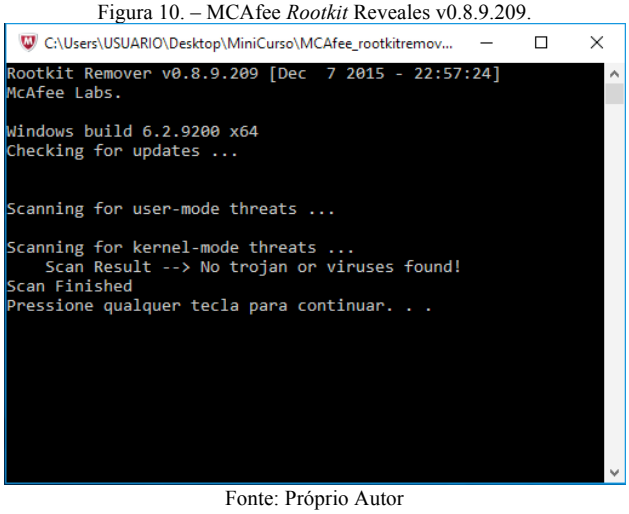
Figura 9. – AVG gmer v2.2.19882.



Fonte: Próprio Autor

O segundo *software* estudado foi o *MCAfee Rootkit Revealer*, ele demonstrou ser bem mais simples e não tão eficiente como o *AVG gmer*, pois ao executar o *software* também como administrador, automaticamente iniciou uma varredura e sem opções de ajustes ou configurações finalizou gerando um relatório da varredura. (ANEXO III).

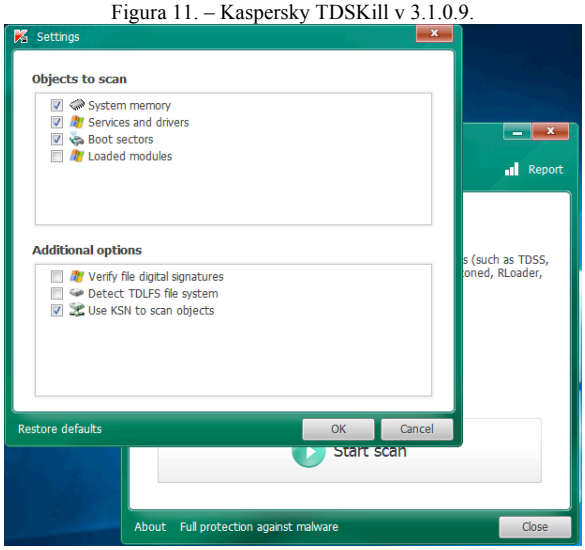
Figura 10. – MCAfee *Rootkit* Reveales v0.8.9.209.



Fonte: Próprio Autor

O terceiro *software* estudado foi o *Kaskersky TDSKill*, também com uma interface bem simples e intuitiva assim que executado como administrador mostra duas opções, para o usuário, a de iniciar a varredura do sistema, ou opções de configuração de varredura.

Figura 11. – Kaspersky TDSKill v 3.1.0.9.



Fonte: Próprio Autor

Além das opções de *scanner* completo do sistema ele ainda fornece uma opção de varredura após a inicialização do Windows, dessa forma executando uma varredura mais profunda do sistema. O relatório gerado pelo *Kaspersky TDSKill* é bem completo, detalhando todos os arquivos verificados, com isso acaba se tornando muito extenso, e o *software* não fornece a opção de exportar o relatório.

Todas as ferramentas testadas executam varreduras para detectar *rootkits*, porém a que se mostrou mais eficiente e mais completa para a tarefa testada foi o *AVG gmer*, apresentando mais ferramentas integradas, e uma variedade maior de opções de varredura do sistema.

8. LAUDO PERICIAL E CADEIA DE CUSTÓDIA

Toda investigação criminal necessita de um laudo da pericia técnica sobre as análises feitas no caso, com os crimes de informática não é diferente, não basta apenas analisar, investigar e obter as respostas, toda investigação forense que envolva computação em seus atos finais necessita da elaboração de um laudo pericial que será entregue ao juiz, e anexado ao processo. “O Laudo Pericial é o documento que relata os

resultados da perícia." (VECCHIA, E. D., 2014). Porém um laudo pericial não descreve apenas os resultados obtidos, ou as respostas das questões que levaram a análise, "nele devem ser descritos a metodologia utilizada, os procedimentos aplicados, os resultados," (VECCHIA, E. D., 2014) e tudo que o perito achar necessário de informação para auxiliar a investigação.

Para um laudo pericial ser criado, o analista forense deve utilizar toda sua capacidade técnica necessária para solucionar as questões levantadas na investigação. "Ele deve saber como "transformar" tudo isso em texto, e em mídias (quando houver anexo em formato digital)" (VECCHIA, E. D., 2014).

Segundo o Autor (VECCHIA, E. D., 2014) existem alguma características mínimas para um laudo pericial de boa qualidade:

- **Clareza**: o laudo deve ser compreensível, acessível, inteligível, pois geralmente os leitores serão juízes e advogados e, não compete a eles a obrigação de dominar assuntos ligados a tecnologia. Para obter a clareza no aludo deve-se evitar a utilização de termos técnicos sempre que possível;
- **Objetividade**: deve-se obter resultado de observação imparcial, independente das preferencias individuais, pois o perito não julga, a função dele é apenas analisar e realizar o processamento de dados, e relatar detalhadamente o que foi encontrado;
- **Itens Indispensáveis:** qual a metodologia utilizada para a analise, relação das evidências, e respostas aos questionamentos (se tiver).

"A documentação confeccionada pelo investigador deve conter a identificação de todos os arquivos coletados (contendo dados como número do caso, nome da pessoa que fez a coleta, data e local) e do armazenamento de cada evidência coletada" (REIS, M. A. 2003). "Além do mais, no tribunal pode-se utilizar um laudo para assegurar a legitimidade das evidências obtidas durante o processo investigativo" (FREITAS, A. R. 2006).

Esse laudo denomina-se cadeia de custódia e é uma peça fundamental no processo de investigação forense. (RODRIGUES, T. S., FOLTRAN, D. C., 2010).

"Não existe um modelo único para todos os institutos ou departamentos de perícia do Brasil (nem no mundo). (VECCHIA, E. D., 2014). Por tanto um laudo pericial ou cadeia de custodia deve conter todas as informações necessárias e solicitadas pela investigação e elaborada de acordo com as normas e exigências da instituição a qual solicitou a análise.

CONCLUSÃO

Com a grande evolução da tecnologia, podemos ver e sentir cada vez mais a dependência no dia a dia. A legislação brasileira aos poucos está se adaptando com a lei nº 12.965/14 de 23 de abril de 2014 (Marco Civil da Internet) e a lei nº 12.737/12 de 02 de dezembro de 2012 art. 154-a (Lei Carolina Dieckmann). Mesmo sendo novas, surge então uma maior necessidade da análise forense computacional para que essas leis tenham o suporte legal em investigações.

Como já visto neste artigo existem várias técnicas e *softwares* maliciosas utilizados por *hacker* e *cracker* para a pratica de golpes cibernéticos, com isso várias empresas investem em ferramentas e técnicas para a tentativa de rastrear os golpes digitais inclusive com *rootkits* ("mecanismos para destruir seus rastros e assegurar sua presença no dispositivo computacional." (VECCHIA, E. D., 2014)). Um perito não depende apenas de ferramentas, mais precisa ter uma grande experiência e conseguir traduzir todas as análises e coletas de provas em uma linguagem de fácil entendimento e com o máximo de conteúdo concreto possível.

Este artigo tratou sobre a análise forense computacional, mostrando várias técnicas e *softwares* maliciosos utilizados em crimes digitais, além de *softwares* para a detecção de *rootkit* em ambiente Windows. Mostrando ainda a grande dificuldade em que os peritos computacionais enfrentam para uma análise e criação de um laudo pericial com base legal e de fácil entendimento em um tribunal.

REFERÊNCIAS BIBLIOGRÁFICAS

BARCELAR R. R., Sistema Operacionais Abertos, 2013, Disponível em: http://www.ricardobarcelar.com.br/aulas/soa/mod1-ger_processos.pdf, Acessado em: 18 de Julho de 2016;

CAMBRIDGE, A. L. **VNC – Virtual Network Computing from AT&T Laboratories**, Cambridge, 2007;

CARVEY, H.; CASEY, E. **Windows Forensic Analysis DVD Toolkit 2ed.** Syngress, 2009;

CERT.br **Centro de estudos, respostas e tratamento de incidentes de segurança no brasil.** Comitê Gestor da Internet no Brasil – CGI.br, disponível em: http://cert.br/, 2007;

CÓDIGO DE PROCESSO PENAL, Disponível em: http://www.planalto.gov.br/ccivil_03/decreto-lei/Del3689Compilado.htm Acessado em: 22 de Abril de 2016.

ELEUTÉRIO, P. M. S.; MACHADO, M. P. **Desvendando a Computação Forense**. Novatec, 2011.

FREITAS, A. R. **Perícia Forense Aplicada à Informática.** Rio de Janeiro: Brasport,. 240 p. 2006.

GALVÃO, R. K. M. **Introdução à Análise Forense em Redes de Computadores: Conceitos, Técnicas e Ferramentas para "Grampos Digitais".** Novatec, 2013.

HOLZ, T. **A Short Visit to the Bot Zoo (malicious bots software).** IEEE Security & Privacy Magazine, 3(3):76–79, 2005;

INMAN, K., RUDIN, N. **Principles and Practice of Criminalistics: The Profession of Forensic Science** (Protocols in Forensic Science) 2000.

KLAUS , S., NELSON, M. **Métodos para detecção local de rootkits e módulos de kernel maliciosos em sistemas Unix.** III Simpósio sobre Segurança da Informação (SSI), São José dos Campos, 2001;

MICROSOFT, **Processes and Threads**. Microsoft Corporation, 2016, disponível em: https://msdn.microsoft.com/en-us/library/windows/desktop/ms684841(v=vs.85).aspx, acessado em: 27 de julho de 2016;

MICROSOFT, **About Processes and Threads**. Microsoft Corporation, 2016, disponível em: https://msdn.microsoft.com/en-us/library/windows/desktop/ms681917(v=vs.85).aspx, acessado em: 27 de julho de 2016;

MICROSOFT, **Microsoft Sysinternals. Microsoft Tecnet.** Microsoft Corporation, 2007 disponível em: http://www.microsoft.com/technet/sysinternals/default.mspx, acessado em: 30 de maio de 2016;

MORENO D. **Introdução ao Pentest.** Novatec, 2015.

NULL A. **Software – Pratcice and experience. Vol. 1,** paginas 201-204, 1971;

PEREIRA, E.; FAGUNDES, L. L.; NEUKAMP, P.; Et. All. **Forense Computacional: fundamentos, tecnologias, e desafios atuais.** VII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Unisinos, 2008;

REIS, M. A. **Forense computacional e sua aplicação em segurança imunológica**. Dissertação de mestrado, Instituto de Computação, Universidade Estadual de Campinas, 2003;

RODRIGUES, T. S., FOLTRAN, D. C. **Análise de ferramentas forense na investigação digital.** Revista de Engenharia e Tecnologia, vol. 2 nº 3, 2010;

SECURITYFOCUS Site da SecurotyFocus, Symantec Corporation, 2007, disponível em:

http://www.securityfocus.com/ , acessado em: 30 de maio de 2016;

SIPIER, J. C.; WARD, B. T.; ROSELLI, G. R. **A United states perspective on the ethical and legal issues of spyware.** In ICEC'05: Proceedings of the 7th international conference on Electronic commerce, paginas 738-743, New York, NY, USA. ACM Press, 2005;

SHUKLA, S.; NAH, F. F. **Web Browsing and spyware intrusion.** Commum. ACM, 48(8):85-90, 2005;

SOPHOS, **Sophos - anti-virus and anti-spam software for businesses.** 2007 Disponível em: http://www.sophos.com/ Acessado em: 10 de Maio de 2016;

SKOUDIS, E.; ZELTSER, L. Malware: **Fighting Malicious Code.** Pentice Hall PTR, 2003;

MCLAUGHLIN, L. **Bot software spreads, causes new worries.** IEEE Distributed Systems Online, 2004 5(6).

UEHARA, M., TSAI, D. – **Evolução dos Microprocessadores Utilizados nos Computadores Pessoais**. Trabalho de Conclusão de Curso , Faculdade de Tecnologia de São Paulo, 2011

VECCHIA, E. D. **Perícia Digital: Da Investigação à Análise Forense**. Millenium, 2014.

VENENA G.; LAVELLA, P.; PICCOLINI, J.; MIRANDA, D.; SANGIRARDI, P.; COELHO, L. **Segurança: Análise Forense.** Escola Superior de Redes RNP, 2009

WEISS, A. **Spyware be gone!** NetWorker, 9(1):18–25, 2005.

ZHANG, Y; PAXSON, V. **Detecting backdoor. In Proc.** 9th USENIX Security Symposium, pag. 157- 170, 2000.

EVIDÊNCIA ELETRÔNICA

FORMULÁRIO DE CADEIA DE CUSTÓDIA

Caso Num.: **Pag.:** **De:**

MÍDIA ELETRÔNICA/DETALHES EQUIPAMENTO

Item:	Descrição:		
Fabricante:	Modelo:	Num. de serie:	

DETALHES SOBRE A IMAGEM DOS DADOS

Data/Hora:	Criada por:	Método usado:	Nome da Imagem:	Partes:
Drive:	HASH:			

CADEIA DE CUSTÓDIA

Destino:	Data/Hora:	Origem:	Destino	Motivo:
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	
	Data:	Nome/Org.:	Nome/Org.:	
	Hora:	Assinatura:	Assinatura:	

ANEXO II

Relatório de Varredura do AVG GMER 2.2.19882

GMER 2.2.19882 - http://www.gmer.net

Rootkit scan 2016-07-27 22:46:58

Windows 6.2.9200 x64 \Device\Harddisk0\DR0 -> \Device\0000001a
VBOX_HARDDISK rev.1.0 30,00GB

Running: AVG gmer.exe; Driver:
C:\Users\USUARIO\AppData\Local\Temp\agadrkow.sys

---- User code sections - GMER 2.2 ----

? C:\Windows\SYSTEM32\iertutil.dll [2816] entry point in ".rdata" section
000000007370ca20

? C:\Windows\SYSTEM32\NTASN1.dll [2816] entry point in ".rdata" section
000000007244bb10

? C:\Windows\system32\apphelp.dll [4424] entry point in ".rdata" section
0000000072370380

---- Threads - GMER 2.2 ----

Thread C:\Windows\system32\csrss.exe [448:4288]
fffff96171924060

---- Registry - GMER 2.2 ----

Reg HKLM\SYSTEM\CurrentControlSet\Control\Session
Manager\kernel\RNG@RNGAuxiliarySeed -1322283330

---- EOF - GMER 2.2 ----

ANEXO III

Relatório de Varredura do MACAfee 0.8.9.209

[TimeStamp: 20160727 224719]Rootkit Remover v0.8.9.209 [Dec 7 2015 - 22:57:24]

McAfee Labs.

Windows build 6.2.9200 x64

Checking for updates ...

Scanning for user-mode threats ...

Scanning for kernel-mode threats ...

Scan Result --> No trojan or viruses found!

Scan Finished