

Procedimentos e Ferramentas para Manutenção de Honeypots de Alta Interatividade

Lucio Henrique Franco^{1*}, Antonio Montes¹

¹Laboratório Associado de Computação e Matemática Aplicada –
Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1758 –
12227-010 São José dos Campos, SP

lucio@lac.inpe.br, montes@lac.inpe.br

Abstract. *As part of honeynets maintenance process, for the monitoring of hostile activities in the Internet, several procedures and tools that automate the high-interaction honeypot management tasks have been developed. Amongst the advantages of the adoption and use of these in honeynets there are the elaboration of procedures documentation, the standardization in the collected material storage, the elimination of errors during honeypot maintenance, the automatization of the tasks and the reduction of the time between the deactivation and re-installation.*

Resumo. *Como parte do processo de manutenção de honeynets, para a monitoração de atividades hostis na Internet, vários procedimentos, padronizações e ferramentas que automatizam as tarefas de gerenciamento de honeypots de alta interatividade vem sendo desenvolvidos. Dentre as vantagens da adoção e uso desses nas honeynets estão a elaboração de documentação dos procedimentos, a padronização no armazenamento do material coletado, a eliminação de erros durante a manutenção dos honeypots, a automatização das tarefas que são executadas e a diminuição do tempo entre a desativação e nova instalação de um host nesta rede.*

1. Introdução

Honeynets são ferramentas de pesquisa que consistem de uma rede projetada especificamente para ser comprometida, elas contêm mecanismos de captura, análise e contenção de tráfego e partem do princípio que todo este tráfego é considerado malicioso [The Honeynet Project, 2001, Spitzner, 2000]. Essas redes são compostas de uma sub-rede administrativa e de vários *hosts* chamados de *honeypots*, que são um recurso de segurança preparado com a finalidade de ser sondado, atacado ou comprometido e registrar essas atividades [Spitzner and Ranum, 2002].

Como parte da operação de um projeto de pesquisa na área de *honeynets* é necessário elaborar e implementar diversos procedimentos para ativar ou desativar um *honeypot* em uma *honeynet*. A padronização destes procedimentos é muito importante, principalmente quando se trabalha com grande número de *honeypots* ou quando várias pessoas do projeto compartilham a tarefa de gerenciá-los.

Estes procedimentos variam conforme a taxonomia dos *honeypots* e vão desde zerrar os discos da máquina, à escolha do sistema operacional que será instalado e os serviços

*Gostaríamos de agradecer em especial aos membros do Projeto Honeynet.BR e a Cláudio Corrêa pelas sugestões e idéias na elaboração dos procedimentos e ferramentas.

que serão disponibilizados, até a configuração de mecanismos para coleta e preservação dos artefatos¹ deixados nos *honeypots* pelos invasores. Os procedimentos ajudam a evitar erros ou esquecimentos ao longo do processo de manutenção dos *honeypots*. Essas falhas podem, por exemplo, contaminar as informações capturadas pelos *honeypots* ou levar o invasor a perceber que está em uma *honeynet*.

Estes procedimentos adotados para o gerenciamento de *honeypots* e as ferramentas para automatizá-los vêm sendo desenvolvidos e aplicados ao longo do tempo desde a implementação das *honeynets*, passando por constantes revisões de aperfeiçoamento.

Este artigo mostrará alguns destes procedimentos e ferramentas desenvolvidos para auxiliar na manutenção dos *honeypots* e ele está dividido da seguinte forma: na Seção 2 serão apresentadas algumas referências à taxonomia dos *honeypots*. Na Seção 3 serão apresentados os procedimentos criados para a manutenção de *honeypots* de alta interatividade. Na Seção 4 serão descritas algumas ferramentas desenvolvidas para automatizar os procedimentos elaborados e empregados nestes *honeypots*. Na 5 serão descritos alguns trabalhos relacionados à manutenção de *honeynets*. Por fim, nas Seções 6 e 7 serão apresentados os trabalhos futuros e as conclusões, respectivamente.

2. Classificação dos Honeypots

Honeypots podem ser classificados como sendo de baixa ou alta interatividade [Spitzner, 2002]. Sistemas de baixa interatividade limitam as ações dos atacantes e coletam poucas informações sobre um ataque, porém, são mais simples de se gerenciar e introduzem pequeno risco ao ambiente de rede visto que o atacante não tem acesso total ao sistema. *Honeypots* de baixa interatividade podem emular serviços de rede como, por exemplo, `ftp`, `http`, entre outros.

Honeypots de alta interatividade são capazes de executar as versões reais dos serviços de rede e permitem que o atacante tenha acesso total à máquina comprometida [Spitzner, 2002]. Esta categoria de *honeypots*, geralmente empregada em *honeynets*, é usada como um recurso para auxiliar no aperfeiçoamento das formas de proteção do ambiente de rede. Estes *honeypots* coletam mais informações que os *honeypots* de baixa interatividade, permitem o acompanhamento dos passos dos invasores e a coleta de artefatos.

Sistemas de baixa interatividade, devido à própria funcionalidade, não devem possibilitar que o atacante tenha controle total do sistema alvo dos ataques e, geralmente, esses sistemas não foram preparados para permitir que o invasor tenha esse controle. Entretanto, falhas nestas aplicações podem levar o invasor a ter acesso ao *honeypot* e comprometer todo o funcionamento da *honeynet* ou da rede de produção em que ele se encontra [Brenton, 2001, Provos, 2004]. Nos *honeypots* de alta interatividade a finalidade é que o invasor possa ter acesso total ao sistema invadido, possibilitando que ele faça *download* de ferramentas e execute comandos. Deste modo, um *honeypot* pode ser utilizado para investigações, para monitoração de acessos não autorizados ou de atividades ilícitas.

Como *honeypots* podem ser utilizados pelos atacantes para desfechar ataques a outras redes, faz-se necessário que as *honeynets* tenham mecanismos de contenção de tráfego de alta granularidade para deter estas atividades.

O projeto HoneyNet.BR [Filho et al., 2002] que fundamenta este trabalho é baseado na Arquitetura GenII descrita por Lance Spitzner [The HoneyNet Project, 2001,

¹Artefatos podem ser definidos como todo o material deixado pelo invasor após o comprometimento de um sistema

Spitzner, 2000] e utiliza *honeypots* de alta interatividade. A gerência desses é complexa, pois não se deve deixar nenhum vestígio da preparação dos sistemas para atuarem como *honeypots* e eles devem conter mecanismos para a captura e preservação dos passos dos atacantes, e para a coleta dos artefatos deixados no sistema. Considerando todos estes aspectos, foram desenvolvidos alguns procedimentos, apresentados nas próximas seções, para auxiliar na manutenção desses *honeypots*, envolvendo o processo de sua configuração inicial, do seu acompanhamento e da sua restauração.

3. Procedimentos Desenvolvidos

Honeynets contêm vários *honeypots* de arquiteturas diferentes, executando diversos sistemas operacionais cada qual provendo serviços locais e de rede. Para a manutenção desses sistemas é necessário desenvolver metodologias e elaborar procedimentos. Estes procedimentos podem ser automatizados por meio de *scripts* que reduzem o tempo entre a desativação e ativação do *honeypot*.

Os procedimentos e *scripts* auxiliam, quando se trabalha com grande número de *hosts* e na execução de passos vitais na configuração de um *honeypot*, como, por exemplo, zerar um disco, eliminando qualquer dado de instalações anteriores ainda contido nele [Farmer and Venema, 2001]. Permitem também a remoção de vestígios da configuração evitando desconfiar o invasor com relação à máquina invadida, o que poderia fazer com que ele não mais retornasse ao sistema, com a possibilidade ainda de anunciar na Internet a localização de tal *honeynet*, afastando os invasores e impedindo a realização de mais pesquisas nesta rede lógica. Outro aspecto importante destes procedimentos é a documentação da configuração do sistema e dos aplicativos de cada *host* que fará parte da *honeynet*, tornando possível um levantamento histórico dos sistemas, serviços e vulnerabilidades que já compuseram a *honeynet* e, também a possibilidade de qualquer integrante do projeto conhecer e executar os passos corretos para se ativar e/ou desativar um *honeypot*.

Na próxima seção são apresentados alguns procedimentos desenvolvidos para a preparação de um *honeypot* de alta interatividade.

3.1. Procedimentos de Instalação dos Honeypots

Para a instalação de um *honeypot* define-se, inicialmente, o sistema operacional, os serviços e suas versões que serão instalados; em seguida é necessária a execução de diversos passos para a configuração desse, conforme a metodologia a ser empregada e definida previamente. Abaixo é apresentada uma visão geral do conjunto de procedimentos básicos que são seguidos durante o processo de instalação de cada *honeypot*:

1. O disco tem seu conteúdo sobrescrito com um padrão constante de dados (usualmente zeros). Este procedimento permite a geração de sua imagem com uma melhor taxa de compressão e facilita a análise pós-invasão, por constarem no disco somente dados referentes à última instalação [Farmer and Venema, 2001];
2. O sistema operacional escolhido para o *honeypot* é instalado, geralmente, utilizando as opções padrões de instalação de cada sistema;
3. Depois de instalado o sistema operacional no *honeypot*, ele tem seus serviços configurados e inicializados;
4. É executada uma seqüência de *scripts* responsáveis pela configuração final do *honeypot*, eles criam usuários e senhas já predefinidos pelo grupo, configuram serviço de sincronização de tempo, serviço de gerenciamento de *logs* para exportar as informações geradas pelas aplicações para um *loghost*, entre outros;
5. Visando o monitoramento dos passos do atacante são configurados no *honeypot* sistemas de captura de teclas que enviam os comandos digitados pelo invasor para

- o *loghost*, através do serviço de *syslog* [The HoneyNet Project, 2001] ou inserindo essas informações diretamente na rede [Barbato and Montes, 2003];
6. São gerados os *hashes* MD5 dos arquivos dos *honeypots* visando armazenar a priori informações de integridade do sistema. Neste passo, são coletadas também informações de status do sistema com a execução de alguns comando, como: *ps*, *netstat*, *lsof*, *socklist*, *df*, entre outros;
 7. É feita uma imagem do disco, que é comprimida e armazenada em outra máquina juntamente com o status do *honeypot* e, posteriormente, esses dados são gravados em uma fita magnética;
 8. Todo o processo é registrado em um livro de registro seguindo o Apêndice A descrito no *HoneyNet Project* [Spitzner, 2001];
 9. O *honeypot* é conectado na *honeynet* e monitorado até o momento da sua retirada da rede.

Outra padronização adotada foi quanto à forma de particionamento dos discos dos *honeypots* e o tamanho dessas partições, pois, no momento da análise forense desses, há sistemas de arquivos que não suportam montar imagens de partições que ultrapassam 2 *Gigabytes*. Um exemplo seria o sistema de arquivo *ext2* do Linux, fazendo necessária a utilização de partições inferiores a 2 *Gigabytes*. Um aprimoramento realizado nesta padronização foi optar-se por trabalhar nos sistemas Unix, quando possível, somente com duas partições: */* e *swap*, pois, *honeypots* com diversas partições como, por exemplo, */*, */home*, */usr*, */var*, *swap*, etc, tornam trabalhoso e demorado o processo de geração da imagem dessas que são criadas uma a uma.

Procedimentos e padronizações também foram desenvolvidos para o armazenamento da imagem após uma instalação e da imagem comprometida do disco de cada *honeypot* juntamente com os status iniciais e finais de cada um. O nome do diretório que contém as imagens e o status de cada sistema é composto pelo nome dado ao *honeypot*, por exemplo, *foobar* mais o nome do sistema operacional instalado nele. Dentro desse diretório raiz são criados diretórios correspondentes ao número de instalações já realizadas neste *honeypot*, o nome para estes diretórios seguem o padrão *install1* para a primeira instalação, *install2*, para a segunda e, assim, sucessivamente. Em cada diretório de instalação são criados mais quatro subdiretórios compostos do prefixo *image* (para as imagens), mais a data atual, mais o nome do *honeypot*, mais o sufixo *orig* para as imagens iniciais ou o sufixo *hack* para as imagens finais deste *honeypot*. O status inicial e o final são armazenados em subdiretórios do mesmo nível do diretório das imagens e tem formação correspondente ao diretório das imagens, porém, o prefixo é *status* e neles são armazenadas as informações extraídas de cada *honeypot* na fase inicial, no seu acompanhamento e na sua desativação. Abaixo um exemplo da estrutura de diretórios criada:

```
/foobar_windows2000server/  
  install1/  
    image_15_03_2004_foobar_orig/  
    image_15_04_2004_foobar_hack/  
    status_15_03_2004_foobar_orig/  
    status_15_04_2004_foobar_hack/  
  install2/  
    image_18_04_2004_foobar_orig/  
    status_19_04_2004_foobar_orig/
```

Estrutura de diretórios criada para armazenamento das imagens e do status dos *honeypots*

Os status gerados pelo acompanhamento dos *honeypots* são armazenados no diretório correspondente ao seu status original, sendo que o nome de cada arquivo é formado pelo prefixo *status* e seguido da data da coleta das informações.

3.2. Procedimentos de Acompanhamento dos Honeypots

Depois que um *honeypot* é disponibilizado na *honeynet* ele passa a ser monitorado e quando invadido os comandos digitados pelos atacantes, suas ações no sistema e outras informações geradas pelas aplicações são enviadas para um *loghost* centralizado e encaminhadas na forma de alertas para os membros do projeto via e-mail, celular ou outro meio que utilize à Internet.

Estes sistemas dos *honeypots* podem vir a sofrer falhas, sejam elas, por problemas de *hardware*, problemas no sistema de arquivos, alguma interrupção de serviço ou do sistema todo devido a um ataque sofrido. Assim, tornou-se necessário implementar um sistema de acompanhamento destes *hosts*. Semanalmente², cada *honeypot* passa pela seguinte série de testes locais via *console*, como mostra a Figura 1.

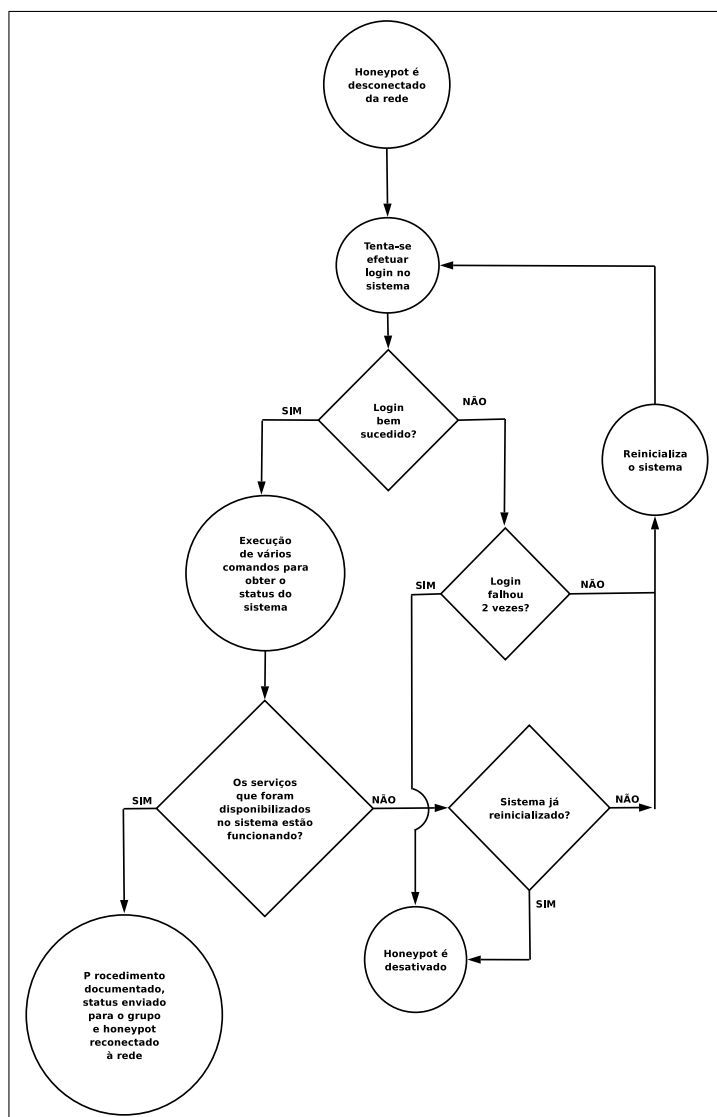


Figura 1: Procedimentos utilizados para o acompanhamento dos *honeypots*.

Ao executar os procedimentos para aferir se um *honeypot* está em funcionamento conforme a configuração inicial é necessário ter alguns cuidados para não deixar vestígios no sistema. Preferencialmente são utilizados binários compilados de forma estática dos comandos executados para a obtenção do status do sistema. Esses binários são executados a partir de um CDROM e têm as saídas de seus comandos direcionados para um disquete.

²Este período é definido pelo grupo

Essas informações são armazenadas, distintamente para cada *honeypot*, em um documento que é enviado por e-mail para todos do grupo. São empregados binários estáticos, pois, esses não fazem uso de bibliotecas e/ou arquivos do sistema, que podem ter sido alterados por algum *rootkit* [Murilo and Steding-Jessen, 2001]. Entretanto, a utilização desses binários estáticos não dará os resultados esperados se o *rootkit* ou outra aplicação instalada no *honeypot*, pelo invasor, fizer uso de módulos de *kernel* para tentar ocultar a invasão. Por outro lado, isso já é conhecido de antemão, uma vez que as atividades do atacante vem sendo monitoradas e a instalação do *rootkit* e seu resultado já foi registrado. Neste caso, o status é extraído do sistema e comparado com anteriores, dependendo da diferença entre eles, devida a influência da instalação de ferramentas no sistema pelo invasor, o grupo pode decidir pela retirada do *honeypot* da *honeynet*.

Na próxima seção são apresentados os procedimentos desenvolvidos e empregados na desativação dos *honeypots*.

3.3. Procedimentos de Desativação dos Honeypots

Depois do *honeypot* ser invadido é tomada a decisão pelo grupo, no momento mais oportuno, de sua retirada da rede. Isso normalmente acontece quando o invasor danifica componentes críticos do sistema ou deixa o *honeypot* num estado inutilizável em decorrência das alterações realizadas no sistema após uma invasão. Para a desativação de cada *honeypot* os procedimentos adotados são semelhantes aos procedimentos de acompanhamento descrito anteriormente. Porém, após a coleta e análise do status do sistema comprometido, os seguintes passos são executados:

1. É feita a imagem da máquina comprometida em outra máquina, armazenando este arquivo junto com a imagem inicial e os status iniciais e finais extraídos do *honeypot*, seguindo a estrutura de diretórios apresentados na seção 3.1;
2. O *host* neste momento é liberado para ser preparado e configurado novamente como um *honeypot*, como descritos na seção 3.1.

Nos *honeypots* em que não é possível realizar *login* no sistema através do *console*, para a geração da imagem de suas partições, em decorrência da invasão sofrida, é realizado o *boot* no *host* através do CDROM ou disquete, utilizando uma distribuição Linux como a `tomsrtbt`³ ou pelo sistema operacional OpenBSD⁴ e, a partir deste ponto, com o auxílio desse outro sistema, as imagens dos discos do *honeypot* são geradas e exportadas para outra máquina.

Na imagem gerada, de cada *honeypot* que foi comprometido, são feitos estudos e análises forenses, comparando-se o status inicial e as modificações feitas pelo invasor no sistema. Posteriormente, estes arquivos juntamente com todas as outras informações de status e acompanhamento do sistema são comprimidas e transferidas para uma fita magnética como forma de *backup*.

Fechando este ciclo de vida do *honeypot* entre sua instalação, configuração, monitoração e desativação, os procedimentos completam e especificam quais tarefas que devem ser e como devem ser executadas em cada estágio do gerenciamento de um *honeypot* para que se tenha um padrão das informações geradas e um armazenamento de forma organizada destes dados coletados e, principalmente, reduzindo o tempo de inatividade de um *honeypot* entre as sua desativação e nova instalação.

Na próxima seção são descritas algumas ferramentas desenvolvidas com a finalidade de automatizar os procedimentos e metodologias citados.

³<http://www.toms.net/rb/>

⁴<http://www.openbsd.org/>

4. Ferramentas Desenvolvidas

Ferramentas empregadas na automatização do processo de configuração de *honeypots* vem sendo desenvolvidas para sistemas Unix, Windows e, principalmente, Linux; elas são desenvolvidas em sua maioria em *shell script* visto que estes dispensam a instalação de aplicações adicionais e devido à sua portabilidade. Alguns desses *scripts* utilizados na fase de instalação dos *honeypots* são descritos abaixo:

4.1. Gerador de Hash MD5 de Arquivos e Compactação dos Arquivos Gerados

A função do *script* apresentado nesta seção é gerar o *hash* MD5⁵ de todos os arquivos num dado diretório; incluindo arquivos de subdiretórios. O sistema pode ter vários diretórios como entrada e ao término de sua execução, as informações resultantes são armazenadas em um arquivo de saída para cada diretório. Por fim, é realizada uma compressão do arquivo gerado.

A formação do nome dos arquivos já comprimidos é composta do nome do *honeypot*, acrescentado do *path* do diretório analisado e do sufixo *md5*. Dentre os diretórios dos sistemas Unix em que esse *script* é executado, pode-se citar: */dev*, */usr/local/etc*, */usr/sbin*, */usr/local/bin*, */usr/bin*, */usr/local/sbin*, */etc*, entre outros. Abaixo um exemplo de vários arquivos gerados pelo *script*:

```
foobar_etc_md5.gz
foobar_usr_bin_md5.gz
foobar_usr_sbin_md5.gz
```

Arquivos compactados gerados pelo *script* contendo o *hash* MD5 dos arquivos dos diretórios de entrada */etc*, */usr/bin* e */usr/sbin*

Este *script* foi desenvolvido para os sistemas Linux, FreeBSD, OpenBSD, Solaris e Windows, esse último utilizando o software *cygwin*⁶. Para distinguir os outros sistemas e para identificar os parâmetros corretamente de cada comando utilizado neste passo, o *script* utiliza o comando `uname -s`, que retorna o sistema operacional em questão, e assim prossegue com a execução para cada sistema correspondente. Casos especiais em sistemas Unix ocorrem, por exemplo, com relação ao diretório */dev* no qual é realizado o *hash* do arquivo que contém a saída do comando `ls -lac` que apresenta a data de alteração do diretório, já que é comum a utilização deste diretório para armazenamento de artefatos.

4.2. Ferramenta para Remoção Segura dos Arquivos Copiados para o Honeypot

Uma ferramenta desenvolvida em *shell script* utilizando, em conjunto, os comandos `awk`, `dd` e `rm`, `find` é responsável por sobrescrever o conteúdo de todos os arquivos de um diretório e seus subdiretórios, dado como entrada, com um padrão de zeros e, posteriormente, removê-los. Isto permite apagar todos os arquivos transferidos para o *honeypot* após sua instalação e que foram necessários para sua configuração, facilitando a remoção de vestígios deixados no sistema durante a sua configuração e facilitando a análise forense desse sistema.

4.3. Configurações Gerais dos Honeypots

Os *scripts* descritos até o momento são integrantes de um sistema geral (*framework*) que segue todos os procedimentos, descritos na seção 3.1, criados para o processo de instalação de um *honeypot*. Este sistema geral foi desenvolvido em módulos o que permite a seleção de quais comando serão executados para determinado *honeypot* segundo a

⁵<http://www.ietf.org/rfc/rfc1321.txt>

⁶<http://www.cygwin.com/>

ordem estabelecida nos procedimentos. Isso possibilita que a configuração de um *honeypot* seja interrompida e que se prossiga, em outra ocasião, a partir do passo subsequente ao que foi interrompido. Abaixo é apresentada a ordem de execução padrão dos diversos *scripts*:

1. **Criação das contas dos usuários:** Todos os *honeypots* têm usuários legítimos que são criados logo após a instalação do sistema. Os usuários têm nomes fictícios e são adicionados através do comando `adduser` com os parâmetros necessários para cada sistema operacional. Nos sistemas Windows esta função é realizada através da janela de gerenciamento de usuário correspondente;
2. **Criação das senhas dos usuários:** Nesta função são criadas as senhas dos usuários adicionados no passo anterior. As senhas seguem um padrão forte especificado pelo grupo, o comando utilizado é o `passwd` com seus parâmetros. Para sistemas Windows a geração de senhas é realizada de forma semelhante ao passo anterior;
3. **Instalação dos mecanismos de coleta de logs:** Como o principal objetivo das *honeynets* é observar as ações dos atacantes, faz-se necessária a utilização nos *honeypots*, de diversos mecanismos para coleta de *logs* [The Honeynet Project, 2001]. Esses são responsáveis por exportar para outra máquina todas as informações sobre o comportamento do sistema que está sendo monitorado. Na arquitetura utilizada, as informações geradas em cada *host* são exportadas para um *loghost*. Este passo estabelece também qual o endereço *IP* do *loghost* e qual a severidade das informações que serão enviadas para ele;
4. **Instalação do mecanismo de sincronização de tempo:** É essencial um sistema para sincronismo de tempo entre os *hosts* para que se possa realizar, de forma correta, a correlação de eventos dos *logs* armazenados. Esta função é responsável por instalar no sistema uma aplicação para ajuste de tempo e suas dependências. Outra configuração realizada neste passo visa especificar o endereço *IP* do servidor com o qual o *honeypot* será sincronizado, este endereço pode ou não estar contido dentro da *honeynet* e deve ser único para toda a rede⁷;
5. **Compilação e instalação do sistema de captura de teclas:** As máquinas Unix da *honeynet* possuem um sistema de captura de teclas que envia o histórico dos comandos digitados via o serviço de `syslog` [The Honeynet Project, 2001] ou módulo de *kernel* [Barbato and Montes, 2003]. Esta função compila e instala este mecanismo;
6. **Remoção dos arquivos fontes instalados:** Todas as aplicações a serem instaladas são baixadas para o *honeypot*. Neste passo são removidos todos esses arquivos fontes que foram transferidos para o *honeypot*, além dos traços deixados na instalação do sistema em arquivos como `$HOME/.bash.history`. O *script* utilizado para realizar tal função é descrito na seção 4.2;
7. **Geração do hash MD5 do sistema:** São gerados os *hashes* MD5 dos arquivos do sistema utilizando o *script* descrito na seção 4.1;
8. **Geração do status do sistema:** Neste momento, após o *honeypot* ser configurado, é gerado o status do sistema com informações de saídas dos comandos `ps`, `df`, `rpcinfo`, `netstat`, `lsof`, entre outros. Essas são utilizadas na análise forense e em comparações com os status gerados durante toda a operação dos *honeypots*. Os arquivos gerados neste passo são enviados por e-mail para os membros do grupo e uma cópia é transferida para uma outra máquina;
9. **Geração da imagem do disco:** Por fim, uma imagem das partições do disco do *honeypot* é transferida para a máquina responsável por armazenar as imagens e

⁷A unicidade deste endereço foi definida pelo grupo

os status desses. O *script* utiliza os comandos `dd` e `netcat` em conjunto. Inicialmente é executado o comando `nc -l -p 10000` na máquina destino para abrir um `server socket TCP` na port 10000 e aguardar conexões do *honeypot*. A saída padrão desse comando é direcionado para um arquivo que será a própria imagem da partição do disco do *honeypot*. No *honeypot* é executado o comando `dd` com os parâmetros para ler blocos de uma dada partição do sistema, em seguida esses são enviados via `pipe` para o comando `netcat`, que atua no *honeypot* como aplicação cliente conectando-se na aplicação servidora na máquina destino e transferindo-os para a máquina repositório de imagens;

10. **Compressão das partições:** A partir de agora o *honeypot* pode ser colocado em operação e a imagem juntamente com seu status inicial são organizados na máquina que armazena essas informações seguindo a estrutura de diretórios descrita na seção 3.1.

Executados estes passos, inicia-se a fase de monitoração do *honeypot*. Na próxima seção será apresentado o *script* de coleta de status de acompanhamento dos *honeypots*.

4.4. Acompanhamento dos Honeypots

Os *honeypots* podem sofrer falhas enquanto estão em produção, estas falhas estão relacionadas a quedas de energia elétrica, falhas no sistema de arquivos, falhas ou interrupção de algum serviço que foi disponibilizado, entre outras. Para tentar manter os *honeypots* o menor tempo possível inativo, semanalmente⁸ cada *host* da *honeynet* passa por uma checagem de diversas funcionalidades como descrita na seção 3.2.

Para automatizar e padronizar este processo de coleta de status dos sistemas Unix⁹, foi desenvolvido um *shell script* que utiliza um CDROM contendo os binários compilados de forma estática e necessários para a realização de tal tarefa e, um disquete que contém o próprio *script* e que também é utilizado para armazenar a saída dos comandos que são executados no sistema do *honeypot*. A execução deste *script* se dá da seguinte forma:

1. Primeiramente, monta-se o disquete no sistema através do comando `mount` com seus respectivos parâmetros;
2. Executa-se o *script*, diretamente do diretório onde o disquete foi montado, passando os parâmetros de localização do CDROM, o diretório onde as informações geradas serão gravadas, entre outros;
3. É feita a verificação e validação da entrada dos parâmetros passados e faz-se a montagem do drive de CDROM no diretório especificado, utilizando-se também o comando `mount`;
4. No próprio *script* está definida a série de comandos, não exaustiva, e seus parâmetros, a serem executados no sistema. Basicamente, são eles¹⁰: `hostname`, `df -h`, `w`, `last`, `netstat -na`, `ps -auwx`, `vmstat`, `rpcinfo -p 0`, `lsof`, `socklist`, `uptime`, `printenv`. Com esta lista de comando já é possível a análise do comportamento do sistema, sendo coletados dados referentes ao sistema de arquivos, memória, serviços e processos que estão sendo executados, portas utilizadas, variáveis de ambiente, entre outros;
5. Posteriormente, é criado o diretório que conterà a saída dos comandos acima e seu nome é constituído da data atual;
6. Os comandos citados são executados um a um, a partir do seu caminho no CDROM, e a saída com seus resultados é acrescentada no final do arquivo de

⁸Este período pode ser variável

⁹Este *script* ainda não foi portado para sistemas Windows, entretanto, nestes sistemas os procedimentos são semelhantes

¹⁰Exemplo de comandos utilizados nos sistemas Linux

status correspondente, o qual está localizado dentro do diretório criado no passo anterior;

7. Por fim, o CDROM e o disquete são desmontados de seus respectivos diretórios.

Abaixo segue um resumo da saída o arquivo gerado do status do *honeypot* chamado *hp1* e que foi armazenado no arquivo `/mnt/floppy/15_03_2004/hp1`.

```
$cat /mnt/floppy/15_03_2004/hp1

### STATIC BINS ###

### hostname ###
hp1

### df -h ###
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/hd0a       1.4G  0.8G  0.6G    75%      /

### w ###
11:07:24 up 3 days, 21:05, 1 user, load average: 0.00, 0.04, 0.36
USER  TTY      FROM    LOGIN@  IDLE   JCPU   PCPU   WHAT
root  tty1    -       8:47am  2:19m  0.35s  0.02s  /bin/sh
```

Arquivo gerado com o status corrente do *honeypot* *hp1*. A saída foi simplificada para fins de exemplificação

Um teste preliminar que vem sendo realizado com esta ferramenta é a extensão da coleta do status utilizando os binários estáticos e os binários do sistema. Possibilitando, desta forma, uma comparação do resultado apresentado entre os comandos dos binários estáticos e a saída dos mesmos no sistema que podem ter sido substituído pelo invasor na instalação de algum *rootkit*, esta observação permite, por exemplo, analisar de forma rápida e sucinta os efeitos de uma tentativa de ocultação de invasão e as características apresentadas nestes tipos de ferramentas utilizadas pelos atacantes.

5. Trabalhos Relacionados

Lance Spitzner [Spitzner, 2002] discutiu a importância de se desenvolver procedimentos para a configuração dos *honeypots* de forma correta e segura conforme cada taxonomia. Porém cada grupo de pesquisa na área de *honeynets* desenvolve suas próprias técnicas para gerenciar seus *honeypots*. Estes grupos compartilham ferramentas e idéias, mas, em nenhum momento foi discutido, nem pela comunidade nem na literatura, particularidades de procedimentos e seus aprimoramentos para a manutenção de um *honeypot* de alta interatividade. Somente com estudos, observações e testes sobre estes sistemas é possível criar e realizar melhorias nos processos de gerenciamento de *honeypots* e *honeynets*.

Há várias ferramentas que podem auxiliar no processo de ativação e desativação de *honeypots*, entre elas o `md5deep`¹¹ que é um programa escrito em linguagem C semelhante ao *script* descrito na seção 4.1 que gera o *hash* MD5 de um determinado diretório. Porém, o *script* realiza todas as funções desta ferramenta e apresenta como vantagem sobre essa, sua alta portabilidade e integração com outros *scripts* desenvolvidos conjuntamente.

Outra ferramenta auxiliar para a remoção segura de arquivos do sistema é o comando `shred` pertencente ao pacote `fileutils`¹² da GNU¹³, porém, com a utilização do *script* desenvolvido com a mesma funcionalidade evita-se a instalação de pacotes extras ou a migração dessas aplicações para outras plataformas, o que acarretaria também, em alguns casos, a mudança de parâmetros e incompatibilidades, um exemplo seria o comando `rm` nos sistemas BSDs e o seu parâmetro `-P` que tem por função sobrescrever diversas vezes um arquivo com os padrões `0xff`, `0x00` e `0xff` antes de removê-lo. Esta funcionalidade ainda não foi implementada, para o mesmo comando, em sistemas Linux.

¹¹<http://md5deep.sourceforge.net/>

¹²<ftp://ftp.gnu.org/gnu/fileutils/>

¹³<http://www.gnu.org/>

6. Trabalhos Futuros

Com a necessidade da realização de testes com diversos sistemas operacionais e aplicações em *honeynets*, os procedimentos elaborados e as ferramentas desenvolvidas para automatizar a configuração e o gerenciamento desses sistemas devem ser revistos para acompanhar as mudanças ou suportar novas características do sistema que está sendo implementado.

Um dos estudos realizados sobre as ferramentas desenvolvidas, concluiu-se que é recomendável a migração da geração do *hash* MD5, dos arquivos do sistema, para SHA1¹⁴ (*Secure Hash Algorithm 1*), devido à saída de 160 bits que este algoritmo utiliza, ao contrário dos 128 bits empregado pelo MD5, evitando colisões nos resultados gerados. Porém, o algoritmo SHA1 não está disponível em todas as distribuições Linux em sua instalação padrão, o que obrigaria a instalação adicional desse software. Como medida de simplificação e compatibilidade das ferramentas com todos os sistemas utilizados na *honeynet*, optou-se por utilizar o algoritmo MD5, que pode ser migrado facilmente para SHA1, quando esse último estiver amplamente disponível.

Dos testes que estão sendo realizados nestas ferramentas, resta validar o *script* de acompanhamento dos *honeypots*, descrito na seção 4.4, quando esse faz uso dos binários do sistema na tentativa de buscar incompatibilidades e diferenças entre os binários estáticos e os binários que podem ter sido substituído pelo invasor. Esta ferramenta ainda mostrará a diferença encontrada entre o resultado da execução destes dois tipos de arquivos e também um histórico de status já coletados anteriormente.

Uma migração que já foi iniciada é a substituição do uso da aplicação *cygwin* nos sistemas Windows para o conjunto de ferramentas GNU¹⁵ de mesma funcionalidade. Esta mudança facilitará a configuração dos *honeypots* que rodam este tipo de sistema já que não haverá a necessidade de instalação de software adicional e, os aplicativos que serão utilizados são de total compatibilidade com o sistema em questão e não mais simulados como ocorre quando se faz uso do *cygwin*.

Com relação a outras ferramentas que serão desenvolvidas, uma delas será a elaboração de um comando *rm* modificado para sistemas Unix. Esse terá a finalidade de retirar uma cópia dos arquivos e/ou diretórios que estão sendo removidos do sistema comprometido. Facilitando, desta maneira, a coleta de artefatos e de alterações do sistema deixados pelos atacantes.

Muitas vezes a recuperação desses elementos removidos é feita através da análise do tráfego de rede capturado. Essa decodificação se torna praticamente impossível quando o atacante faz uso de criptografia para a transferência de seus dados. Com o novo comando, os arquivos e/ou diretórios que seriam apagados do sistema, passarão a ser movidos para outro diretório oculto no próprio *host* ou enviado para o *loghost*, de modo transparente para o invasor. Esse estará certo que seus artefatos foram removidos do sistema, quando na verdade, este material estará sendo movido para um local desconhecido do invasor e armazenado para posterior análise.

As ferramentas utilizadas atualmente também poderão ser migradas para outras linguagens como a linguagem *Perl* ou a linguagem *C* caso haja necessidade.

7. Conclusão

A criação de metodologias, elaboração de procedimentos e o desenvolvimento de ferramentas para a configuração de um *honeypot* de alta interatividade é de grande relevância,

¹⁴<http://www.ietf.org/rfc/rfc3174.txt>

¹⁵<http://unxutils.sourceforge.net/>

porque automatizam processos e auxiliam na coleta de informações. São criados padrões a serem seguidos por todo o grupo, evitando erros como desligar o *honeypot* sem a geração do seu status final, o que prejudicaria a comparação das características do sistema no momento de sua instalação com o seu status depois de uma invasão. Diminuindo também o tempo que o *honeypot* fica fora da rede até sua próxima configuração.

As ferramentas vem sendo desenvolvidas e empregadas ao longo do tempo desde a implantação do projeto HoneyNet.BR em dezembro de 2001, quando tiveram início as definições dos procedimentos para se configurar todos os *hosts* que fariam parte desta rede. O acompanhamento dessas *honeynets* desde então mostrou a utilidade do desenvolvimento destas ferramentas para automatizar a configuração de um *honeypot*, seus ajustes e refinamentos.

Referências

- Barbato, L. G. C. and Montes, A. (2003). SMaRT - Session Monitoring and Replay Tool. In *Trabalho em Segurança de Redes (GTS'02.2003)*, Rio de Janeiro, RJ. <http://www.honeynet.org.br/papers/smart-gts2003.pdf>.
- Brenton, C. (2001). Honeynets. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4232. ISSN 0277-786X CODEN: PSISDG.
- Farmer, D. and Venema, W. (2001). Being Prepared for Intrusion. *Dr. Dobb's Journal*, 26(4).
- Filho, A. B., Amaral, A. S. M. S., Montes, A., Hoepers, C., Steding-Jessen, K., Franco, L. H., and Chaves, M. H. P. C. (2002). HoneyNet.BR: Desenvolvimento e Implantação de um Sistema para Avaliação de Atividades Hostis na Internet Brasileira. In *Anais do IV Simpósio sobre Segurança em Informática (SSI'2002)*, pages 19–25, São José dos Campos, SP. <http://www.lac.inpe.br/security/honeynet/papers/hnbr-ssi2002.pdf>.
- Murilo, N. and Steding-Jessen, K. (2001). Métodos para Detecção Local de Rootkits e Módulos de Kernel Maliciosos em Sistemas Unix. In *Anais do III Simpósio sobre Segurança em Informática (SSI'2001)*, pages 133–139, São José dos Campos, SP.
- Provos, N. (2004). Honeyd Security Advisory 2004-001. <http://www.honeyd.org/adv.2004-01.asc>.
- Spitzner, L. (2000). Learning the Tools and the Tactics of the Enemy with Honeynets. In *Proceedings of the 12th Annual Computer Security Incident Handling Conference*, Chicago, IL, USA. <http://www.first.org/events/progconf/2000/D3-09.pdf>.
- Spitzner, L. (2001). HoneyPot Deployment Log. <http://project.honeynet.org/alliance/AppendixA.txt>.
- Spitzner, L. (2002). HOSUS(HoneyPot Surveillance System). In *login: Magazine of Usenix and Sage*, volume 27. <http://www.usenix.org/publications/login/2002-12/pdfs/spitzner.pdf>.
- Spitzner, L. and Ranum, M. (2002). HoneyPots: Tracking Hackers. In *SANS 2002 Annual Conference*, Orlando, Florida, USA.
- The HoneyNet Project (2001). *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community*. Addison-Wesley, 1st edition. ISBN 0-201-74613-1.