Implementando Segurança e Controle em Redes de Computadores

Leandro Márcio Bertholdo *Liane M. R. Tarouco*

Instituto de Informática
Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Campus do Vale
91509-900 Porto Alegre, RS
E-mail:
berthold@penta.ufrgs.br liane@penta.ufrgs.br

Resumo

Este trabalho apresenta o sistema CUCO[1], uma proposta de uma nova filosofia para segurança de sistemas. O sistema CUCO se utiliza de conceitos de gerenciamento de redes como SNMPv2, e segurança como DES e MD5, aliados a implementação de um conjunto de ferramentas que garantam a integridade dos sistemas envolvidos. O resultado é um sistema que alerta sobre tentativas de ataques e situações de risco que as diversas máquinas sob seu controle estão expostas, aumentando desta forma o nível de controle que o administrador tem sobre a segurança da sua rede.

Abstract

This works describes the CUCO[2] system, a proposal for a new filosophy in system security. The CUCO system harness network management concepts like SNMPv2 and security concepts like DES and MD5, associated with a set of tools that ensure the integrity of systems. The result is a system that alerts on attack attempts and risk situations to which the machines under its control that the administrator has on the security of its network.

1. Introdução

Após alguns anos conectada à Internet, a universidade começou a perceber que, juntamente com os benefícios de estar diretamente ligada ao meio científico mundial, também estava se expondo a um ambiente que se tornava a cada dia mais hostil. Esta abertura têm colocado a instituição em uma posição delicada, onde freqüentes problemas relativos a segurança de informações e operacionabilidade das máquinas tem mostrado a fragilidade ante as investidas realizadas.

A comunidade mundial vem se defendendo como pode. Inúmeras ferramentas são construídas anualmente, muitas delas, como SATAN [VEN95] [BER96], ISS [CER93], PINGWARE [BEL96] e AutoHack [MUF95] com o intuito de realizar uma auditoria sobre a segurança de máquinas ou subredes, outras visando detectar uma invasão ocorrida no passado e algumas tentando garantir a segurança através de filtros, como por exemplo o TCPWrapper [VEN92]. Uma outra forma de manter a segurança de um site, que também vem sendo usado é o próprio isolamento das máquinas da Internet, como são as várias técnicas de *Firewall* [CHE94] [CHA95].

Após exaustivos estudos e acompanhamentos sobre as vulnerabilidades existentes e as formas de explorá-las, pressentiu-se dentre tudo o que foi divulgado no meio científico a falta de um dispositivo que consiga captar o momento exato de uma invasão. Informação esta, extremamente útil para que o administrador possa tomar as medidas necessárias de forma eficiente, e em um momento crucial. Em situações práticas, tem-se por várias

[1] Analogia ao mecanismo de relógios de parede, que notifica um evento: as mudanças de hora.
[2] Analogy to wallclock mechanism, that notify an event: hour changes.

vezes notado, que um reconhecimento tardio de uma invasão (após uma ou duas horas) pode tornar a situação fora de controle e comprometer inúmeros outros *sites* de forma irremediável e não rastreável.

A criação do sistema CUCO tem por principal objetivo preencher esta lacuna existente, possibilitando alertar o administrador local sobre a identificação de alterações que possam de alguma forma comprometer a segurança do sistema. Ele tem como meta possibilitar ao administrador tomar conhecimento de possíveis tentativas de violação da segurança, além de tentar descobrir qual foi a técnica usada em uma determinada invasão e possibilitar algumas atitudes emergenciais de forma a impedir que o intruso tome posse do sistema; fornecendo assim o tempo necessário para que uma determinada falha na segurança seja sanada.

Inicialmente, o sistema foi projetado para identificar e alertar o administrador responsável pela segurança da rede em casos de comprometimento ou suspeita de invasão. Entre os problemas escolhidos a serem alertados cita-se:

- Alguma das máquinas teve seus sistema operacional ou aplicativos comprometidos por ferramentas com *rootkit*[3].
- Algum host está coletando senhas na rede com algum programa do tipo *sniffer*.
- Algum host registrou acesso a partir de algum local desconhecido.
- Algum arquivo de configuração do sistema foi alterado, como por exemplo o próprio arquivo de senhas.
- Houve algum registro considerado "atípico" nos logs do sistema.

Além dos itens expostos acima, o sistema é ágil e de fácil utilização, sem que ele necessite despender maiores cuidados do administrador para o conjunto das máquinas monitoradas. A aplicação também possibilita ao seu usuário obter informações de forma rápida e consistente no caso da ocorrência de uma invasão. O objetivo desta informação é tentar identificar a origem e o responsável por um possível ataque.

2. O Sistema CUCO

Visando solucionar paleativamente o problema de um sistema aberto, diretamente conectado a uma rede de computadores, surgiram pesquisas que visam a detecção automatizada de intrusões[4]. Estes mecanismos, totalmente automáticos tentam precisar tentativas de violação na segurança em uma determinada máquina, e, desta forma, revelar novas falhas como a exploração de novos bugs, estouros de pilha e outros. Existem alguns desenvolvimentos nesta área que se utilizam de técnicas de inteligência artificial, e que hoje abrangem basicamente dois grupos[SPA93]:

- detecção de anomalias;
- detecção de violação.

O sistema CUCO utiliza a primeira abordagem. A detecção por anomalia é baseada na premissa que atividade intrusa freqüentemente se manifesta como uma anormalidade. Esta abordagem se baseia em constantes medidas realizadas pelo sistema, de forma a detectar através de grandes variâncias nestas métricas a indicação de uma intrusão. Um exemplo disso é um grande número de conexões sendo rejeitadas pelo sistema por *time-out* (uma possível evidência de um ataque por negação de serviço - synflooding).

Para detectar uma anomalia, o sistema mantém uma base de conhecimento sobre os diferentes sistemas operacionais e usuários. Este conhecimento é utilizado para parametrizar o estado atual de uma determinada máquina. Isto será melhor explicado nos capítulos a seguir.

A detecção de intrusão é uma abordagem nova de segurança. Ela provê uma sensação de segurança em uma rede de computadores e dados, enquanto permite a eles operarem de uma forma "aberta". A meta desta técnica é identificar, preferivelmente em tempo real, o uso não autorizado, mau uso, ou abuso de sistemas de computação, tanto por usuários internos como invasores externos [MUK94].

A segunda abordagem, detecção de violação ou abuso, baseia-se em técnicas específicas de representação de conhecimento sobre "comportamentos inaceitáveis", e tentativas de detectar estas ocorrências. Um exemplo disso seria a detecção do abuso do fingerd e sendmail, usado no ataque do "Internet Worm" [SPA88].

---

[3] O *RootKit* é um conjunto de ferramentas utilizados por hackers que substituem diversos aplicativos como z2, es, fix, sl, ic, ps, ns, ls e outros de forma a facilitar a entrada no sistema e de esconder os traços das invasões.

[4] Conceitualmente, intrusão é compreendido como a detecção de atividade ilegal e aquisição de privilégios que não pode ser detectado através do fluxo normal de informações e os modelos de controle de acesso

2.1 A Estrutura do Sistema CUCO

A estrutura de funcionamento da aplicação usa uma filosofia de monitoração, onde um único gerente monitora via SNMPv2 as métricas de dezenas de hosts (Figura 1).

Cada host possui um agente SNMPv2, que se comunica com o gerente, repassando a ele possíveis alertas de violações ou respondendo a consultas sobre o estado de seus objetos. A consulta é uma forma de verificação de atividade do agente. A simples falha neste processo pode significar o um ataque está em andamento, ou seja, toda a segurança acrescentada ao sistema depende basicamente da comunicação confiável entre os hosts monitorados em o programa gerente rodando na máquina diretora.

A estrutura da aplicação é formada por dois módulos básicos, que implementam além do protocolo de gerência, outras funções:

Módulo Monitor: Composto por vários programas de aplicação, responsáveis por identificar uma possível invasão.

Módulo Diretor ou Gerente: Composto por uma interface amigável que permite realizar consultas SNMP e ser capaz de notificar uma anomalia qualquer para o administrador responsável pela segurança (oficial de segurança) do *site*, para que este tome as devidas providências.

A troca de mensagens entre os módulos gerente e agente é realizada através do protocolo SNMPv2. Esta escolha é devido as suas características incorporadas a partir do protocolo S-SNMP, que garante o sigilo e autenticidade das mensagens trocadas entre os parceiros.

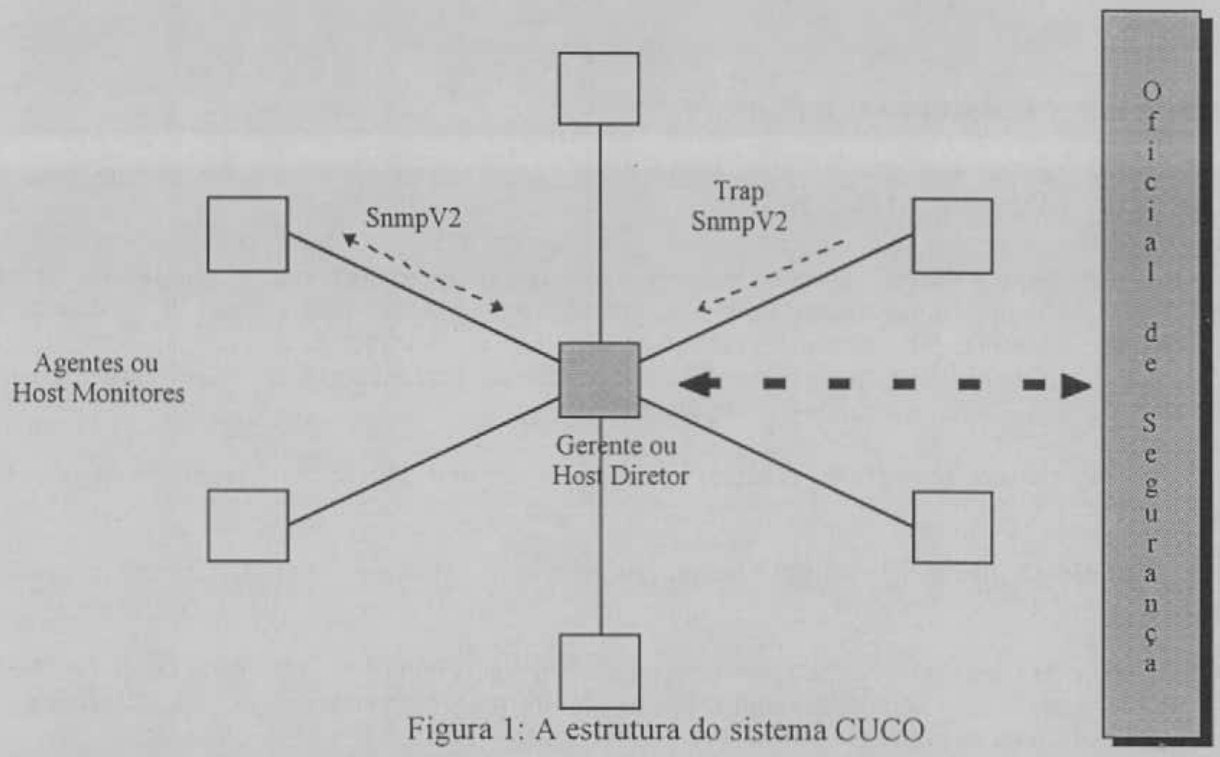

Figura 1: A estrutura do sistema CUCO

2.2 O Módulo Monitor

O módulo monitor é o coração do sistema CUCO, ele é o responsável direto pela segurança do host onde ele está instalado. Cabe a ele detectar qualquer tipo de ataque ou modificações em arquivos de configuração ou mesmo instalação de *trap-doors* [GAR94] tanto no sistema operacional quanto em aplicativos. O monitor garante a integridade de dados sensíveis do usuário, além de manter uma base de informações (MIB) sobre o estado dos objetos, acionando traps quando necessário. Existem cinco atribuições para o módulo monitor, e que estão assim distribuídas:

1. A implementação dos diversos módulos capazes de reconhecer a ocorrência de uma invasão, tais qual um monitor para as conexões, verificadores de integridade do sistema operacional e das configurações do sistema e um controlador dos estados das interfaces de rede dos hosts.

2. Um analisador automatizado para os logs do sistema, visando assim obter um maior controle sobre o estado das máquinas. Isto é realizado através de programas que automaticamente vasculham os logs existentes nas máquinas a procura de indícios.

3. A implementação do protocolo de gerência, o agente SNMPv2.

4. A implementação de um módulo capaz de prover informações, a partir da máquina atacada.

5. E, por último, um módulo responsável por recolher maiores informações sobre atividades suspeitas que estejam sendo realizadas por um usuário do sistema. Este módulo deve ser composto de ferramentas que implementam modificações nos códigos do programa shell e de um monitor a nível de terminal de usuário (tty), além de vários scripts que permitam extrair maiores informações do sistema. Este módulo foi parcialmente implementado, pois já existe na bibliografia aplicações que implementam monitores de tty [BELa] e shells modificados, necessitando apenas adaptações para funcionar diretamente com a interface do sistema CUCO.

A figura abaixo mostra o inter-relacionamento de todos os submódulos existentes, e que operam em conjunto para permitir que as funções supra citadas sejam realizadas.

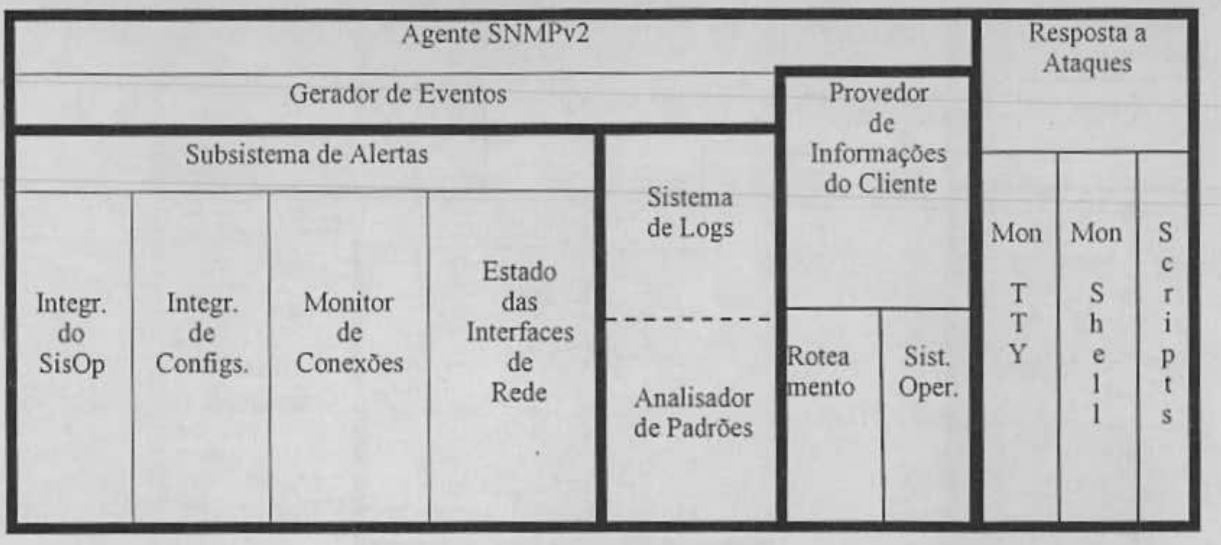

Figura 2: A estrutura do módulo monitor

2.2.1 O Subsistema de Alertas

Este subsistema é composto por um conjunto de procedimentos que garantem a integridade das configurações e do sistema operacional, além de um monitor de conexões e um verificador de interfaces de rede. Todas estas funções constituem o real sistema de alertas da aplicação CUCO.

Os alertas gerados por este subsistema baseiam-se em verificações periódicas dos arquivos de configuração e binários do sistema operacional, além de uma visão geral de como se encontra o estado do sistema em determinados instantes.

2.2.1.1 Os Procedimentos de Avaliação de Integridade

A avaliação de Integridade, tanto do Sistema operacional quanto das configurações do sistema são realizadas utilizando o algoritmo MD5. Esta avaliação é realizada em dois níveis, um considerado obrigatório (binários do sistema operacional), e outro opcional (diversos arquivos de configuração e/ou dados). Os resultados obtidos com a execução do algoritmo alimentam diretamente os diversos objetos da MIB SNMP. Periodicamente o gerente consulta os dados de todos os agentes sob sua jurisdição, para uma confrontação com uma base de dados existente no host diretor (Figura 3).

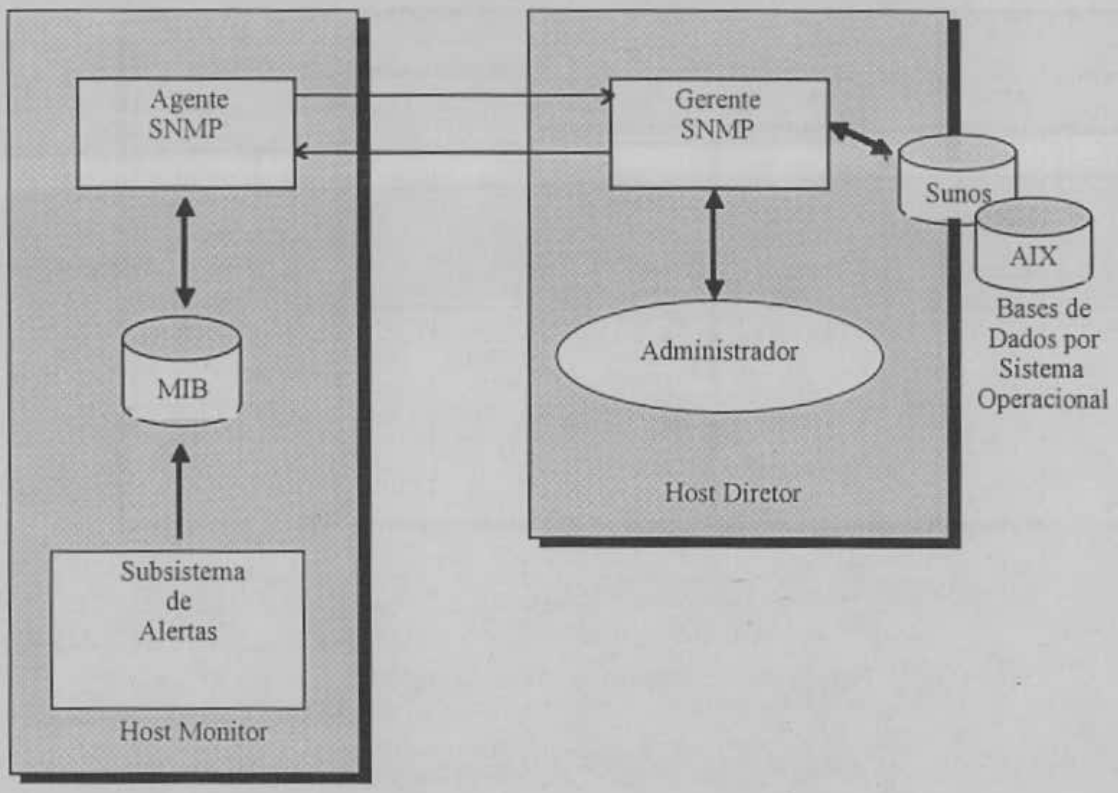

Figura 3: Os módulos de Integridade

No caso da verificação de integridade do Sistema Operacional, o gerente compara o valor dos objetos da MIB fornecidos pelo agente, com uma base de dados local, onde se encontram os MD5 de algumas aplicações consideradas vitais pelo sistema Unix. Se a verificação for referente a um arquivo de configuração do sistema, será consultada uma base de dados organizada por host monitorado, onde lá se encontra um índice com o MD5 de todos os arquivos de configuração considerados importantes, além de uma cópia destes arquivos, para que se verifique qual alteração foi realizada.

A manutenção das cópias dos arquivos de configuração não é um procedimento automático do sistema, e cabe ao administrador mantê-los atualizados. Este processo pode ser razoavelmente automatizado, mas isto é feito ao próprio custo e risco do administrador. Apesar de parecer inviável a primeira vista, este procedimento não é tão penoso, e é de grande valia. Nos dois locais onde o sistema foi testado, mudanças de configurações são incomuns, e, no caso de uma invasão, são de grande utilidade para a verificação de alterações[5].
A implementação atual do protótipo possui alguns programas binários escolhidos para estarem sob constante monitoração (Tabela 1), além de alguns arquivos de configuração do sistema (Tabela 2).

SunOs 4.1.3	AIX 4.1.4
/bin/login	/usr/bin/login
/usr/kvm/ps	/usr/bin/ps
/usr/etc/in.telnetd	/usr/sbin/telnetd
/usr/etc/in.ftpd	/usr/sbin/ftpd
/usr/etc/in.rshd	/usr/sbin/rshd
/usr/etc/in.rlogind	/usr/sbin/rlogind
/usr/etc/inetd	/usr/sbin/inetd
/usr/bin/passwd	/usr/bin/passwd
/usr/bin/su	/usr/bin/su
/usr/etc/ifconfig	/etc/ifconfig

Tabela 1: Binários monitorados pelo sistema

[5] Nem sempre alterações realizadas em arquivos de configuração do sistema são facilmente perceptíveis; como por exemplo a remoção de um único caracter, que pode deixar o site totalmente desprotegido. Este tipo de sutileza foi observado em uma das invasões descobertas.

SunOs 4.1.3	AIX 4.1.4
/etc/passwd	/etc/passwd
/etc/rc.local	/etc/rc.tcpip
/etc/hosts.equiv	/etc/hosts.equiv
/etc/hosts.allow	/etc/hosts.allow
/etc/hosts.deny	/etc/hosts/deny
/etc/inetd.conf	/etc/inetd.conf
/etc/syslog.conf	/etc/syslog.conf
/etc/sendmail.cf	/etc/sendmail.cf
/etc/services	/etc/services
/etc/aliases	/etc/aliases
$HOME/.forward	$HOME/.forward
$HOME/.rhosts	$HOME/.rhosts

Tabela 2: Arquivos de configuração monitorados pelo sistema

Objeto a ser gerenciado	Interpretação
/etc/hosts.equiv $HOME/.rhosts	Alterações permitem acesso ao sistema sem autenticação
/etc/ifconfig	Possibilidade de ocultar *eavesdropper* na rede
/vmunix ou correlato	Alteração no kernel do sistema para instalação de trap-doors
/etc/passwd	Usuário com privilégios especiais, ou facilitação para acesso indevido
rc.local, inetd.conf, services ou correlatos	Alteração no tipo de serviço de rede disponibilizado
/etc/syslog.conf	Tentativa de ocultar a passagem pelo sistema
Arquivos de diretórios /bin /usr/bin /sbin	Possibilidade de instalação de trap-doors

Tabela 3: Alguns eventos capazes de gerar alarmes

Acima estão algumas das verificações realizadas pelo sistema (Tabela 3). Logicamente existem outras que são inerentes, tais como o agente responsável pelas verificações e as suas tabelas de dados, ambas inseridas no código da aplicação, o qual é averiguado sempre que o programa for executado. Todas as verificações realizadas pelo sistema baseiam-se em funções *hashing*, mais especificamente no algoritmo MD5[SCH94]. A implementação do subsistema de alertas possui diferentes níveis de configuração. Existe um nível mínimo, um máximo e um customizável pelo usuário. Estes níveis referem-se basicamente a quais arquivos se estenderá o reconhecimento de autenticidade.

2.2.1.2 O Monitor de Conexões do Sistema

O Monitor de conexões do sistema é também um dos componentes do subsistema de alertas. Através dele são identificadas, em uma primeira proposição, ameaças como SYN flooding. A informação sobre o estado das conexões TCP já é um objeto provido pela MIB SNMP.

O objeto da MIB analisado é TcpConnEntry, que fornece uma visão geral das conexões TCP, e o estado e que cada uma se encontra. Caso haja mais de cinco pedidos de conexões enfileiradas, esperando para serem atendidas (estado synReceived(4)) , o sistema CUCO considera como um possível ataque em andamento contra o sistema (Figura 4).

Informações sobre o estado das conexões são de vital importância para que se confirmem suspeitas a respeito de invasões que estejam ocorrendo, ou mesmo que o host em questão esteja sendo utilizado como trampolim para realizar ataques contra outros sites. Dentre as informações obtidas por este tipo de consultas, estão inclusas: o

número da porta em cada lado da conexão e o endereço IP das duas partes envolvidas, além do número de bytes disponíveis nos vetores de envio e recepção de mensagens. O sistema CUCO analisa estas informações através de um script em PERL e repassa, no caso de alguma suspeita, as informações obtidas ao administrador de segurança da rede.

```
Active Internet connections (including servers)
Prot Recv-Q Send-Q Local Addr Foreign Address       (state)
tcp  0      44  circulo.nntp  jukebox.plug-in..1051 ESTABLISHD
tcp  0       0  circulo.nntp  lakesis.fapesp.b.2524 ESTABLISHD
tcp  0      80  circulo.telne porta3.tche.br.1024   ESTABLISHD
tcp  0       0  *.*           *.*                   CLOSED
tcp  0       0  *.smtp        *.*                   LISTEN
tcp  0       0  *.writesrv    *.*                   LISTEN
tcp  0       0  *.smux        *.*                   LISTEN
tcp  0       0  *.nntp        *.*                   LISTEN
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
tcp  0       0  circulo.www   circulo.pop-rs.r.2342 SYN RECV
```

Figura 4: Conexões de uma máquina sob ataque ao serviço WWW

A análise deste comando permite a detecção de tentativas de ataque utilizando técnicas de SYN flooding. Dependendo da implementação que está sendo usada pelo agressor, o programa monitor pode localizar a sua origem e gerar um relatório sobre a máquina que está sendo usada como fonte do ataque, e, se possível, obter informações a respeito da pessoa que está realizando esta agressão e outras informações como o responsável pelo site ou pessoa de contato.

2.2.1.3 O Controlador de Estado das Interfaces de Rede

O controle sobre o estado das interfaces de rede, é tido como de fundamental importância para o sistema CUCO. Esta verificação, por si só evita que maiores danos sejam causados por hackers que consigam burlar algum host situado em ponto crítico da rede, ou seja, em um dos principais backbones. O objetivo primordial deste subsistema é evitar que se consiga realizar uma monitoração do tráfego da rede (*eavversdropper*), obtendo assim, nomes e senhas de usuários em outras máquinas, situadas em pontos mais bem protegidos da rede, ou mesmo em outros sites.

O sistema CUCO realiza este controle através da manutenção de um objeto experimental na base de dados SNMP. O objeto criado chama-se "ifStatus" é um complemento ao objeto "ifAdminStatus", já existente na MIB. Este novo objeto é mais detalhado, possuindo uma conjunto de valores diferentes do existente hoje. Ele permite uma melhor especificação do estado de uma determinada interface. Neste novo conjunto de valores válidos para o objeto se inclui broadcast(4), debug(5), e principalmente promiscuous(6). Através dele o host diretor pode avaliar se existe algo suspeito ocorrendo em determinada estação. Esta opção precisa ser desabilitada em hosts que utilizam agentes RMON ou rodam programas que executam monitorações na rede, como TCPDUMP, caso contrário, estas máquinas sempre estarão sendo consideradas como "sob ataque" pelo sistema CUCO.

A avaliação sobre o estado das interfaces é realizada por uma função denominada CheckStatus( ) (Figura 5). Esta função usa os mesmos recursos que a comando ifconfig usa para obter uma lista das interfaces existentes em cada host e o seu estado. É através desta função que o objeto ifStatus da MIB SNMP é mantido.

```
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
        inet 143.54.1.20 netmask ffffff00 broadcast 143.54.1.0
lo0: flags=49<UP,LOOPBACK,RUNNING>
        inet 127.0.0.1 netmask ff000000
```

Figura 5: Informação sobre o estado das interfaces obtido por CheckStatus()

2.2.2 O Sistema de Logs e o Analisador de Padrões

Os logs de atividades do sistema são em geral informações preciosas para qualquer administrador que saiba e deseje fazer bom uso delas. O módulo de análise de dados envolve um conjunto de programas responsáveis pela geração e manuseio das informações providas pelo sistema operacional e aplicações, tais como httpd, telnetd, ftpd, logind, rlogind, rshd, tcpd e xtacacsd e outros que o administrador deseje ter sua saída analisada.

Este subsistema baseia-se na procura de mensagens padrões do sistema, como os da figura 6, além de muitas outras que venham a revelar indícios de tentativa de subverter um determinado host, ou que não sejam consideradas pelo administrador como rotineiras. É interessante salientar que o sistema notificará todas as mensagens não consideradas como inofensivas pelo administrador. Este módulo pode ser utilizado independentemente do módulo de gerenciamento.

Este submódulo foi projetado para automaticamente executar e verificar os arquivos gerados pelo sistema de logs, localizando violações ou atividades não usuais realizadas contra o sistema. O analisador de logs utiliza para isso uma base de dados de regras de filtragem para os logs gerados pelo sistema operacional e aplicações.

O código da aplicação é subdividido em dois programas, que agem de forma idêntica: um que possibilita uma análise geral dos logs do sistema, gerando assim um relatório sobre todas as "infrações" que foram registradas nas diferentes aplicações; e outro responsável pela análise simultânea dos logs gerados em cada host monitorado.

A base de dados de regras é formada basicamente por três arquivos: um para identificar o que é considerado uma violação pelo sistema (log.violação) (Figura 6), outro para identificar possíveis avisos importantes registrados pelo sistema, como reboots, daemons reestartados, etc. (log.warning), e ainda outro que especifica tudo o que deve ser ignorado pelo sistema (log.ignore) (Figura 7).

No arquivo log.violação estão registrados padrões verificados que certificam mais de 90% de chances de uma tentativa de hacking do sistema, como pode ser verificado no trecho do arquivo abaixo.

```
WIZ
DEBUG
UUDECODE
VRFY decode
VRFY uudecode
VRFY lp
VRFY demo
VRFY guest
EXPN root
inetd.conf
```

Figura 6: Trecho do arquivo log.violação

No arquivo de log.warning são especificadas quaisquer entradas de log que merecem uma atenção especial do administrador do sistema. Esta foi a idéia principal que gerou este submódulo: reduzir a quantidade de log a ser inspecionada pelo administrador.

```
reject
rshd
REFUSED
rexec
illegal
ILLEGAL
-ERR Password
!=
SITE EXEC
```

Figura 7: Trecho do arquivo log.warning

Conforme a configuração das facilities do arquivo syslog.conf utilizada nos testes, os arquivos de log do sistema costumam tornan-se excessivamente extensos, registrando todas as informações possíveis. Sendo assim, é necessário um procedimento que identifique mensagens através das quais pode-se reconhecer uma possível violações realizada em aplicações (ftp, sendmail), de mensagens comuns de monitoração do sistema, como confirmações de entrega de mensagens, conexões realizadas com sucesso, etc. O arquivo log.ignore é utilizado para identificar estas mensagens "sem importância", das desconhecidas do administrador.

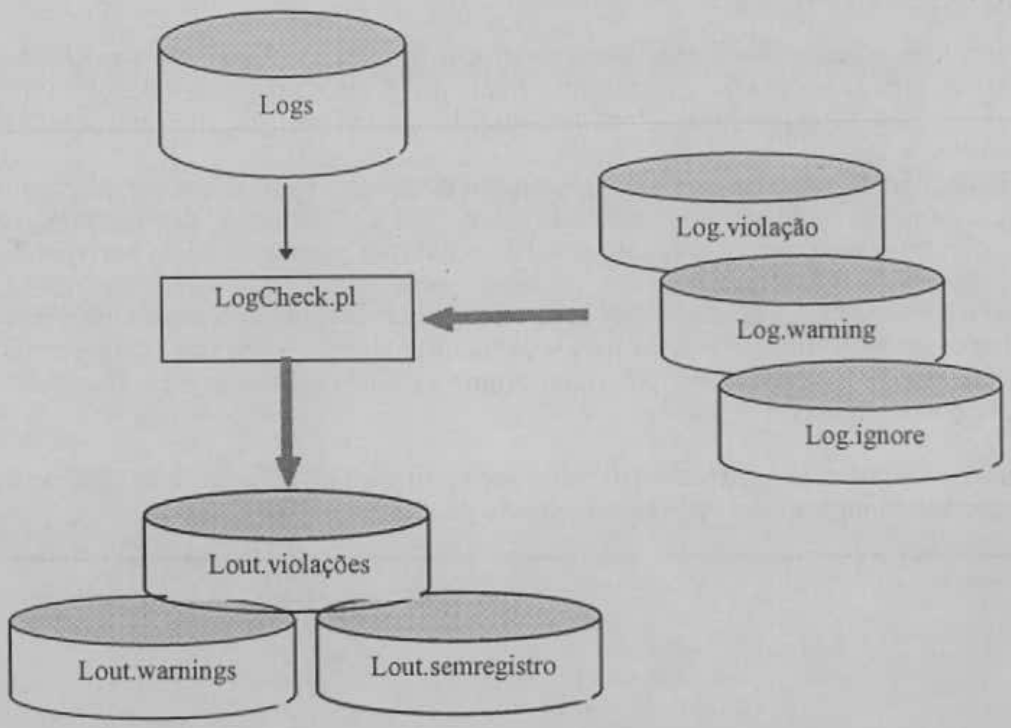

Figura 8: A estrutura do analisador de logs

O programa logcheck.pl (Figura 8), processa os logs do sistema de uma forma completa, ou seja, todos os dados que forem encontrados nos arquivos de logs configurados (diretório /var/log), são inspecionados. Como resultado são gerados três relatórios a respeito de suas conclusões:

- Lout.violações: Todas as tentativas de violação, segundo as regras configuradas em log.violações, ou seja, sabidamente consideradas como um ataque. A existência deste arquivo após o processamento dos logs indica que algum host monitor, ou o próprio host diretor está, ou tem ficado recentemente, sob alguma forma de ataque.
- Lout.warning: Todas as informações de log relevantes que o administrador deve inspecionar diariamente, ou em periodos regulares curtos.
- Lout.semregistro: Relato de entradas desconhecidas ou inesperadas nos logs do sistema. No tempo de adaptação do sistema cuco à sua rede este tipo de mensagens são comuns, necessitando ser revisado freqüentemente, para encaixar tais padrões em algum dos arquivos de regras existentes.

Em sua segunda forma (AnLog.pl) (Figura 9), o analisador de logs é configurado no host monitor utilizando uma chamada ao sistema (tail -f $logfile) para monitorar em tempo real as ocorrências repassadas pelo daemon syslogd. Esta forma de obter os logs foi escolhida pela sua simplicidade e portabilidade, em relações a outros testes onde foram propostas mudanças em programas como syslog, ou mesmo o próprio syslogd. A sua desvantagem é que o sistema fica vulnerável a um ataque ao daemon syslogd.

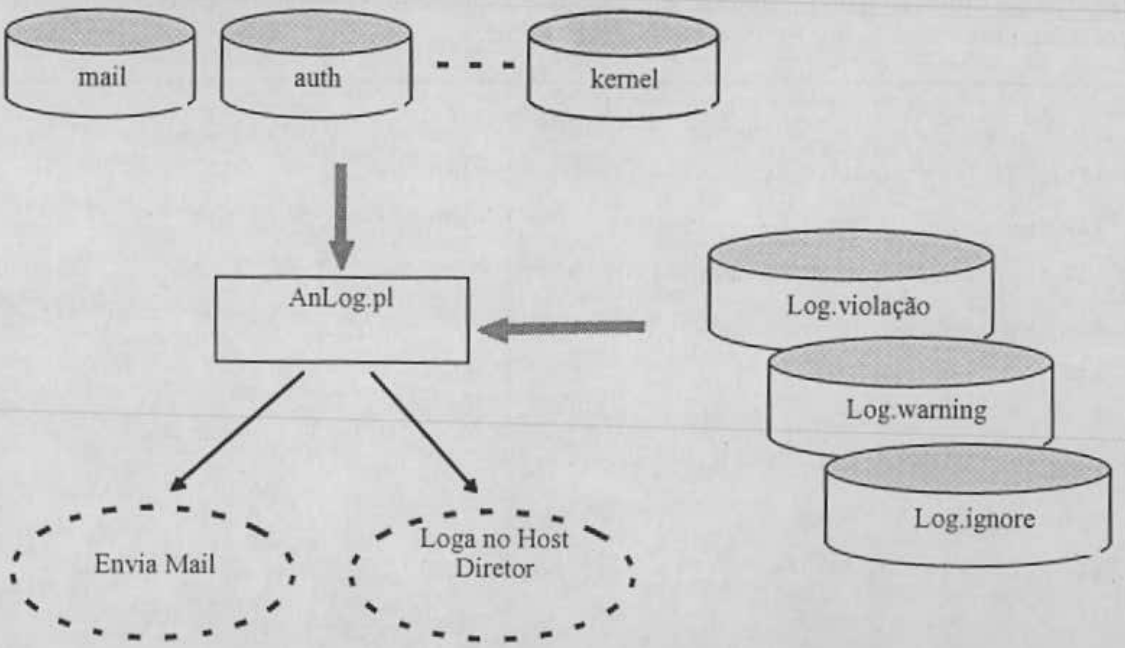

Figura 9: O analisador de logs no host monitor

O analisador de logs no host monitor possui uma forma diferente de registrar os problemas relatados ao host diretor. A forma mais comumente usada hoje em dia seria a de centralizar todos os logs gerados por todas as máquina, o que geraria um tráfego dispensável na rede, além de superlotar o sistema de arquivos do host diretor e exigir um esforço computacional multiplicado várias vezes. Em solução a isto, optou-se pela criação do programa AnLog.pl que executa no host monitor, e registra, via o próprio syslogd (local7) do host diretor somente as entradas que julgue necessárias (regras dos arquivo log.warning e log.semregistro). Uma segunda opção do sistema é a de dar um tratamento especial para os casos que combinem com as regras descritas no log.violação: enviar um mail para o administrador responsável por aquele host monitor (Figura 10).

```
From root Fri Jan 10 14:40:26 1997
Received: from penta.ufrgs.br (penta.ufrgs.br [143.54.1.20]) by circulo.pop-
rs.rnp.br (8.7.5/8.7.3) with SMTP id OAA16072 for <berthold@circulo.pop-
rs.rnp.br>; Fri, 10 Jan 1997 14:40:25 -0300
From: cuco@penta.ufrgs.br
Received: from circulo.pop-rs.rnp.br (circulo.pop-rs.rnp.br [200.132.0.21])
by penta.ufrgs.br (051895/8.6.11) with SMTP id OAA14330 for
berthold@circulo.pop-rs.rnp.br; Fri, 10 Jan 1997 14:40:55 -0300
Date: Fri, 10 Jan 1997 14:40:55 -0300
Message-Id: <199701101740.OAA14330@penta.ufrgs.br>
To: berthold
Subject: CUCO::ALERTA

*****************************
mail: Nov 22 16:28:04 penta.ufrgs.br sendmail[5911]: "wiz" command from
minuano.inf.ufrgs.br (143.54.7.18)
*****************************
```

Figura 10: Exemplo de mail para o administrador

Este submódulo do sistema CUCO também considera dois tipos de logs: os redirecionáveis, gerados por aplicações como TcpWrapper e sendmail, que se utilizam a facilidade syslog; e aqueles gerados por programas como /bin/login, que são inerentemente locais.

O sistema CUCO possui 2 programas **wcheck** e **ucheck**, usados para garantir a integridade de arquivos de auditoria como /etc/wtmp e /var/log/utmp, facilmente adulteráveis por aplicações como "zap", que vem em ferramentas como rootkit. A Figura 11 registrada as saídas dos programas **wcheck** e **ucheck**, criadas exatamente para garantir a integridade destes arquivos de auditoria. As entradas com o comentário "<NULL?>", foram resultado da exclusão do usuário "teste", utilizando uma implementação do programa **wedit.c**, uma implementação semelhante a utilizada pelo programa "zap". Consultas realizadas a estes arquivos por aplicações tradicionais (last, por exemplo) não são capazes de notar tais diferenças.

```
user[     ftp] line[ ftp9568] host[mithrandir.ibase]
user[        ] line[        ] host[                ] <NULL?>
user[        ] line[        ] host[                ] <NULL?>
user[ aluno96] line[   ttyp2] host[minuano.inf.ufrg]
user[        ] line[   ttyp2] host[                ]
user[        ] line[   ttyp0] host[                ]
user[lisianeh] line[   ttyp0] host[ tigrao.ufrgs.br]
user[     ftp] line[ftp10007] host[  200.255.253.23]
user[        ] line[ftp10007] host[                ]
user[        ] line[ftp10491] host[                ]
user[        ] line[        ] host[                ] <NULL?>
user[lisianeh] line[ftp10522] host[ tigrao.ufrgs.br]
user[        ] line[ftp10522] host[                ]
user[  lauren] line[   ttyp5] host[     noc.tche.br]
user[berthold] line[   ttyp5] host[  porta2.tche.br]
user[        ] line[   ttyp4] host[                ]
user[berthold] line[   ttyp4] host[  porta2.tche.br]
user[berthold] line[ftp10809] host[         circulo]
user[        ] line[ftp10809] host[                ]
user[berthold] line[ftp10810] host[  porta2.tche.br]
user[        ] line[ftp10810] host[                ]
```

Figura 11: Saída do programa wcheck para SunOs 4.1.3

2.3 O Módulo Diretor ou Gerenciador

O módulo gerente por sua vez é composto por três partes básicas (Figura 12):

1. A interface do sistema, toda baseada em hiperdocumentos, usando um navegador bem conhecido como o Netscape; um servidor para o protocolo HTTP implementando algumas diretivas básicas do protocolo; programas geradores de código misto HTML e JAVA scripts [FLA96].

2. Um gerente SNMPv2, criado para negociar com os vários agentes remotos. A sua implementação utilizam o pacote CMU.

3. Conjunto de programas C, scripts PERL [WAL96] e scripts SHELL, capazes de analisar e obter informações sobre uma possível fonte de ataque, verificar o funcionamento ou não dos hosts sobre monitoração e realizar analises de dados em geral.

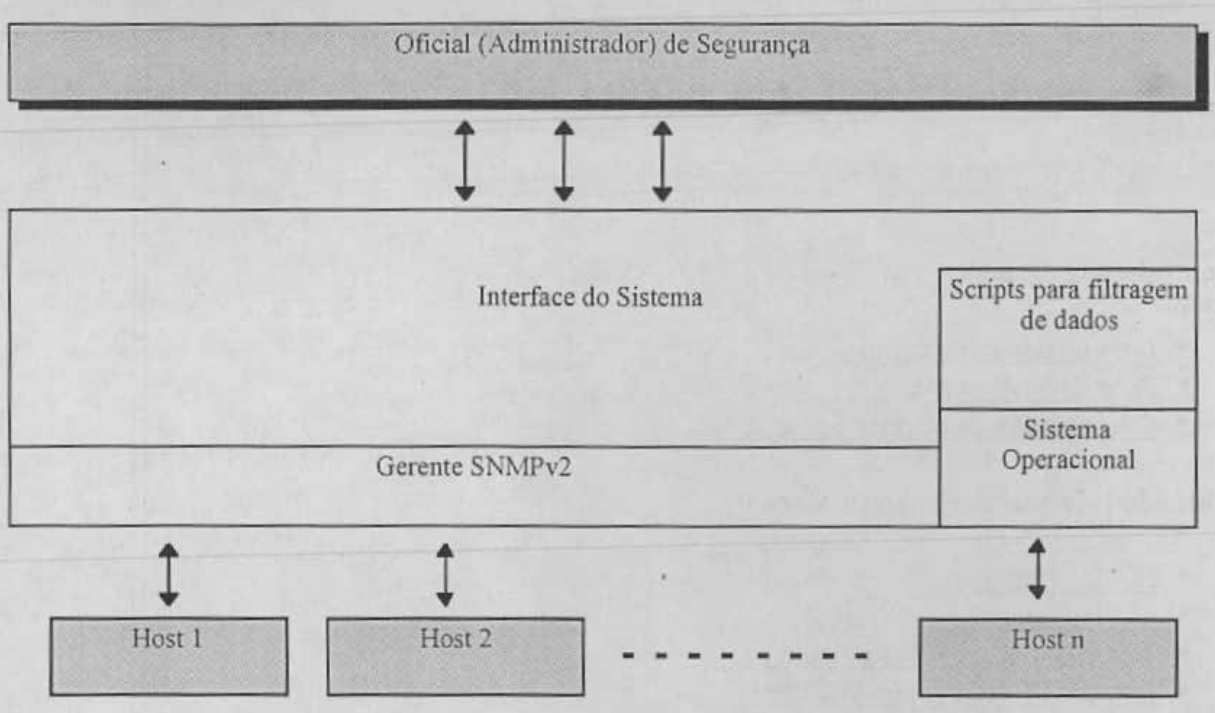

Figura 12: A estrutura do módulo diretor

2.3.1 A Interface do Sistema

Sendo parte do objetivo do sistema CUCO a sua utilização em uma variedades de máquinas situadas no campus da universidade, a portabilidade da aplicação para as diferentes plataformas existentes é um ponto fundamental. Durante a concepção do sistema, a portabilidade e conseqüente simplicidade foi sempre levada em consideração[6]. Em face disso, optou-se por construir a interface do sistema utilizando uma estratégia que está se consagrando dia a dia: uma interface utilizando navegadores (browsers) multimídia (Figura 13). A interface possui scripts JAVA [FLA96] embebidos na linguagem HTML, permitindo desta forma acessar os dados gerados pelos diferentes programas de aplicação e pelo gerente SNMP.

[6] Sempre foi levado em consideração princípios básicos de segurança na construção do código da aplicação, princípios como KISS (*Keep it Simple Stupid)*

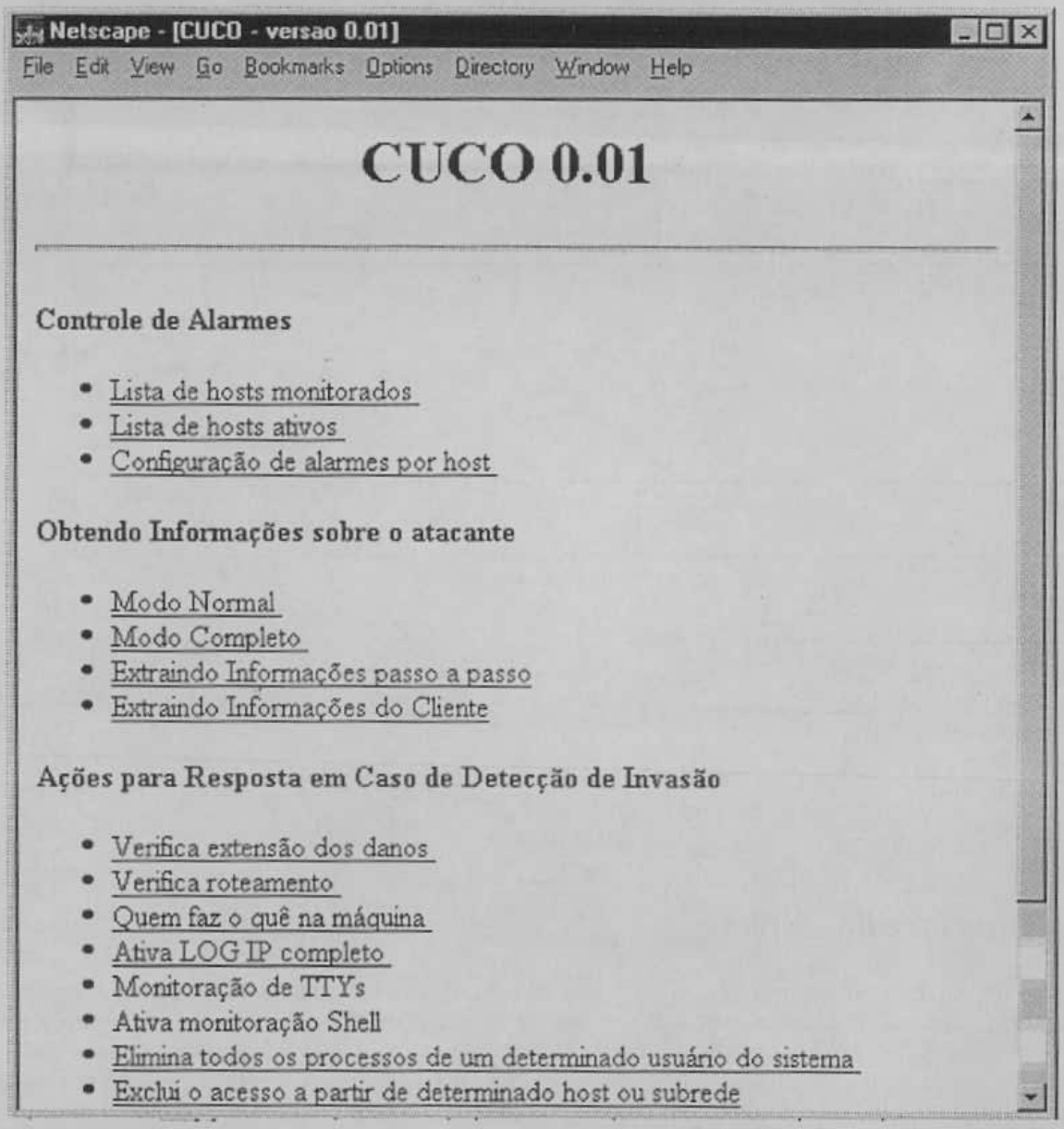

Figura 13: A interface do sistema CUCO

O sistema é inicialmente disparado por programa chamado "cuco". A partir deste programa, o processo inicial é dividido via chamada ao sistema (fork()) em dois: um responsável pela interface direta com o usuário, realizada através do browser Netscape ou Mosaic; outro responsável pelo controle do servidor HTTP. Os dois processos compartilham um mesmo número de porta de comunicação gerado anteriormente ao fork de processos. Desta forma a comunicação entre o cliente e o servidor HTTP é mantida.

2.3.2 Localizando a Fonte do ataque

Um dos desafios para detecção de uma intrusão em um ambiente de rede é sem dúvida o de seguir as pistas deixadas pelo agressor, sejam elas baseadas em uma máquina ou uma pessoa que tenha originado o ataque, ou mesmo em um arquivo deixado pelo agressor. Deve-se ter sempre em mente que esta origem pode se mover por diversos sites da Internet durante o transcorrer de um ataque (ataque sincronizado). Por exemplo, um invasor pode usar diferentes contas de usuários em diferentes máquinas no transcorrer de um único ataque, visando com isso dissuadir o administrador da rede e ocultar a verdadeira fonte do ataque.

O sistema CUCO possui um módulo exclusivo para detecção do agressor. Este subsistema coleta informações de várias fontes independentes, inclusive o próprio estado da rede e o seu comportamento no momento do ataque. Todas estas informações são registradas em um arquivo e posteriormente exibidas ao administrador encarregado da segurança da rede. A identificação de um possível agressor pode ser realizadas pelo responsável pela segurança da rede de duas formas diferentes, segundo o seu critério:

• Um modo **normal**: realiza buscas padrão para descoberta e investigação de um usuário em uma determinado host. Entenda-se padrão como aquelas normalmente realizadas por um administrador de rede em sua tentativa de obter conhecimento sobre um determinado usuário ou máquina da Internet, como nslookup, traceroute, whois, etc.

• Um modo **agressivo**: Este modo de operação, mais completo que o anterior, procura descobrir a qualquer custo a origem de uma determinada conexão, além de possibilitar um reconhecimento das máquinas situadas nos arredores do host agressor. O aplicativo leva em conta as diferentes respostas que o sistema pode ter obtido a partir dos métodos anteriores, recomendando ao administrador que tente obter maiores informações sobre o domínio em questão através das diversas máquinas que se encontram entre ele e a máquina agressora, bastando simplesmente uma seleção do mouse pelo administrador, para que um novo host seja investigado. A busca de informações neste modo estende-se até um completo *port-scanning* do host em questão.

2.4 O Protocolo de Comunicação

O protocolo escolhido para implementar a comunicação e gerenciamento de informações foi SNMPv2 [STA93], por prover um nível aceitável de confiabilidade. O protocolo SNMPv1 não é utilizado, pois embora largamente difundido em um esquema de gerenciamento de rede, possui falhas inerentes de segurança. Falhas como a incapacidade para autenticar a fonte da mensagem bem como prevenir a sua interceptação.

Mesmo incorporando algumas características do S-SNMP, o SNMPv2 ainda possui imperfeições intrínsecas ao protocolo sobre o qual trabalha, no caso UDP. Entretanto, considerando as necessidades de segurança na comunicação entre o agente e o gerente, e, apesar de ainda possuir algunas falhas, SNMPv2 se adapta bem ao problema e conseguem certificar:

- Integridade dos dados: Elimina problemas como inserção, duplicação e ressequenciamento de mensagens através de um esquema de sincronismo e *time-stamp*.
- Autenticação da origem: Provê confirmação da fonte de uma mensagem usando um esquema de chave pública.
- Confidencialidade dos dados: Provê confidencialidade através de criptografia usando um algoritmo simétrico (DES).

Dentre os diversos modos de operação do SNMPv2, optou-se pelo modo "privado e autenticado", que supre todas as características acima.

A seguir está um trecho da MIB SNMP que foi criada para atender às necessidades do sistema CUCO

```
hostDiretor OBJECT-TYPE
              SYNTAX  DisplayString (SIZE (0..255))
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
                  "Informa o host diretor ao qual deve se
                   reportar e ter como único gerente do
                   sistema CUCO."
              ::= { cuco 1 }
hostId OBJECT-TYPE
              SYNTAX  DisplayString (SIZE (0..255))
              ACCESS  read-write
              STATUS  mandatory
              DESCRIPTION
                  "Informa um host que deve perder acesso ao
                   sistema. As conexões atuais são terminadas
                   e novos pedidos são indeferidos. Somente
                   utilizado se o sistema CUCO estiver com o
                   módulo de resposta a ataques habilitado."
              ::= { cuco 2 }

userId OBJECT-TYPE
              SYNTAX  DisplayString (SIZE (0..255))
              ACCESS  read-write
```

```
                STATUS  mandatory
                DESCRIPTION
                    "Informa um usuário a ser eliminado do
                     sistema. Somente utilizado se o sistema
                     CUCO estiver com módulo de resposta a
                     ataques habilitado."
                ::= { cuco 3 }

ativaTcpDump OBJECT-TYPE
                SYNTAX  INTEGER {
                        inativo(0),
                        ativo(1),
                        }

                ACCESS  read-write
                STATUS  mandatory
                DESCRIPTION
                    "Ativa dump de pacotes. Somente utilizado
                     se o sistema CUCO estiver com o módulo
                     de resposta a ataques habilitado."
                ::= { cuco 4 }


            ifNumber OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                    "Número de interfaces de rede do sistema"
                ::= { cuco 5 1 }

            ifTable OBJECT-TYPE
                SYNTAX  SEQUENCE OF IfEntry
                ACCESS  not-accessible
                STATUS  mandatory
                DESCRIPTION
                    "Uma lista de interfaces. O número de
                     entradas é dado por ifNumber."
                := { cuco 5 2 }

            ifEntry OBJECT-TYPE
                SYNTAX  IfEntry
                ACCESS  not-accessible
                STATUS  mandatory
                DESCRIPTION
                    "Cada interface contem mais alguns objetos"
                INDEX   { ifIndex }
                          ::= { ifTable 1 }

            IfEntry ::=  SEQUENCE {
                  ifIndex  INTEGER,
                  ifDescr  DisplayString,
                  ifStatus INTEGER,
                  }

            ifIndex OBJECT-TYPE
                SYNTAX  INTEGER
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                    "Um valor único para cada interface, com
                     valor variando entre 1 e ifNumber."
                ::= { ifEntry 1 }

            ifDescr OBJECT-TYPE
                SYNTAX  DisplayString (SIZE (0..255))
                ACCESS  read-only
                STATUS  mandatory
```

```
        DESCRIPTION
            "Descrição textual da interface."
        ::= { ifEntry 2 }

    ifStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    up(1),
                    down(2),
                    testing(3)
                    broadcast(4)
                    debug(5)
                    promiscuous(6)
                  }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "O estado da interface, segundo os valores
             retornados pela função ifstatus()"
        ::= { ifEntry 3 }
```

Figura 14: Trecho da MIB experimental criada

3. CONCLUSÕES

Tendo em vista o número crescente de incidentes e a falta de uma ferramenta que aborde de forma completa um problema específico como a segurança de máquinas em um ambiente sabidamente vulnerável, a estratégia de monitoração proposta é de grande valia. O sistema CUCO vem preencher uma lacuna deixada entre ferramentas como SATAN, para verificação pró-ativa de segurança e outras como firewalls e filtros, que solucionam estes mesmo problema através de restrições ao acesso. Esta ferramenta é primeira a incorporar características como alarmes e gerenciamento remoto de segurança através de SNMPv2, além de possuir características de análise de dados e monitoração de usuários e processos. A ferramenta possui um grande potencial, podendo inclusive isolar e detectar novas formas de ataque aos protocolos e serviços hoje disponíveis via Internet.

Durante o tempo em que a ferramenta esteve em desenvolvimento, inúmeras tentativas de acesso indevido ou de tentativas de explorar bugs antigos em aplicações como sendmail, foram detectadas. Em simulações de ataques realizados contra o sistema, a aplicação teve, no seu pior caso, apenas o registro de modificação de arquivos. Nesta simulação havia sido explorado um estouro de pilha. A aplicação alertou o ataque quando houve a tentativa de instalar um sniffer e uma trapdoor na máquina. Isto evitou que o sistema fosse adulterado após a violação, mas não foi possível reconhecer através dos logs gerados pelo sistema, qual aplicação foi explorada.

Em outras simulações, onde o nosso arquivo de passwords foi transportado (ação comum realizada por hackers), o sistema registrou a saída deste arquivo via logs do sistema e relatou imediatamente. Isto tornou possível a localização da máquina agressora, bem como a identificação do usuário que a estava ocupando no momento, via contato com o administrador do site agressor.

O sistema CUCO inegavelmente aumentou a tranqüilidade com a qual problemas de segurança são tratados nas máquinas monitoradas. Ele prove uma garantia básica: o sistema operacional está intacto, tanto em seus arquivos quanto em sua configuração. As máquinas ainda estão vulneráveis a aplicações como sendmail, mas temos a garantia que mais cedo ou mais tarde as fontes dos ataques serão descobertas.

Dentre as possibilidades de expansões no uso da ferramenta, estão previstas melhorias tais quais um monitor para terminais de usuários (tty) e um monitor para aplicações como sh e csh, adaptando de ferramentas já existentes.

REFERÊNCIAS

[BEL96] BELLCORE. **PINGWARE System Master Log Summary.** Http://www.belcore.com/demotoo/SECURITY/pingmaster.html, 1996

[BER96] BERTHOLDO, Leandro; TAROUCO, Liane. **Uma Análise do Software de Segurança SATAN - Security Administrator Tool for Analyzing Networks**. II WAIS Fortaleza - CE, 18-20 de Maio de 1996, pp 149-160.

[CER93] CERT/CC. **Internet Security Scanner (ISS).** Computer Emergence Response Team Advisory 93:14, September, 1993.

[CHA95] CHAPMAN, Brent; ZWICKY, Elizabeth. **Building Internet Firewalls**. O'Reilly & Associates, 1995.

[CHE94] CHESWICK, William; BELLOVIN, Steven. **Firewalls and Internet Security: Repelling the Wily Hacker**. Quarta edição. Addison Wesley Publishing Company, 1994.

[FLA96] FLANAGAN, David; **Java in a Nutshell. A Desktop Quick Reference for Java Programmers.** O'Reilly & Associates, Inc.. First Edition, February, 1996.

[GAR94] GARFINKEL, Simson; SPAFFORD Gene. **Practical Unix Security.** O'Reilly & Associates, Inc. 1991. Sétima edição 1994.

[MUF95] MUFFETT, Alec; **WAN-hacking with AutoHack - Auditing security *behind* the firewall**. Sun Microsystems United Kingdom. The Fifth USENIX UNIX Security Symposium, Salt Lake City, Utah. June 5-7, 1995.

[MUK94] MUKHERJEE, Biswanath; HEBERLEIN, Todd L.; LEVITT, Karl N.; **Network Intrusion Detection.** IEEE Network, May/June 1994, p.26-41.

[SCH94] SCHNEIDER, B. **Applied Cryptography**. John Wiley & Sons, 1994.

[SPA88] SPAFFORD, Eugene H.; **The Internet Worm Program: An Analysis.** Departament of Computer Sciences, Purdue University. Purdue Technical Report CSD-TR-823. November, 1998. Ftp://ftp.cs.purdue.edu.

[STA93] W.Stallings. **SNMP, SNMPv2 and CMIP - The Practical Guide to Network-Managment Standards**, Addison-Wesley, 1993.

[VEN92] VENEMA, Wietse. **TCP WRAPPER: Network Monitoring, Access Control, and Booby Traps.** Mathematics and computing Science - Eindhoven University of Technology. In proceedings of the Third USENIX Security Symposium, Baltimore, Maryland. September 1992.

[VEN95] VENEMA, W.. **S.A.T.A.N. A Summary. Internet Security.** March 1995.

[WAL96] WALL, L.. **PERL Reference Manual - version 5.002beta1g.** January 1996.