IMPROVING STATEFUL INSPECTION LOG ANALYSIS
MELHORANDO A ANÁLISE DE *LOGS* COM INSPEÇÃO DE ESTADO

Cristiano Lincoln Mattos (lincoln@cesar.org.br)
Centro de Estudos e Sistemas Avançados do Recife - CESAR
Universidade Federal de Pernambuco – CIn/UFPE
Tempest Security Technologies

Fabio Silva (fabio@cesar.org.br)
Centro de Estudos e Sistemas Avançados do Recife - CESAR
Universidade Federal de Pernambuco – CIn/UFPE

Evandro Curvelo Hora (evandro@cesar.org.br)
Centro de Estudos e Sistemas Avançados do Recife - CESAR
Universidade Federal de Pernambuco – CIn/UFPE
Universidade Federal de Sergipe – DCCE/UFS
Tempest Security Technologies

Marco Antônio Carnut (kiko@cesar.org.br)
Centro de Estudos e Sistemas Avançados do Recife - CESAR
Universidade Federal de Pernambuco – CIn/UFPE
Tempest Security Technologies

ABSTRACT

This paper presents a method for analyzing firewall log files that recognizes related connections from application level protocols, much like "stateful inspection" firewalls such as Linux's IPTables or Checkpoint's Firewall-1 do for allowing/denying traffic. Both a general framework and specific examples are discussed, and analysis results from sample data. Implications and potential for security applications are also presented.

RESUMO

Este artigo apresenta um método para análise de logs *de* firewalls *que reconhecem conexões correlatas de protocolos de nível de aplicação, da mesma forma que* firewalls *com inspeção de estado, tais como o IPTables do Linux ou o Firewall-1 da Checkpoint o fazem para permitir/barrar tráfego. Tanto um apanhado geral quanto exemplos específicos são apresentados, bem como resultados de análises de dados-exemplo. Implicações e potencial para aplicações de segurança também são apresentados.*

1 INTRODUCTION
INTRODUÇÃO

Log analysis is an often forgotten activity, perhaps because of the massive amount of logs generated by firewalls and the lack of good automated tools to aid in their analysis and interpretation. Most analysis tools limit themselves to tally up connection counts; since many application protocols originate and/or receive several connections, this is neither a convenient nor didactic way of presenting the results of such an analysis.

A análise de *logs* é uma atividade freqüentemente esquecida, talvez por causa da maciça quantidade de *logs* gerados pelos *firewalls* e a falta de boas ferramentas automatizadas para auxiliar na sua análise e interpretação. A maioria das ferramentas se limita a tabular contagens de conexões; dado que muitos protocolos de aplicação originam e/ou recebem diversas conexões, essa não é uma maneira nem conveniente nem didática de apresentar os resultados de tais análises.

It would be much better if the analysis tool could identify related connections from common application protocols, tallying them separately or ignoring them. The resulting report would be much more readable, allowing for easier identification of normal and anomalous behavior.

Seria muito melhor se a ferramenta de análise pudesse identificar conexões correlatas de protocolos de aplicação comuns, tabulando-as separadamente ou ignorando-as. O relatório resultante seria muito mais legível, permitindo uma identificação mais fácil do comportamento normal e do anômalo.

This paper is organized as follows: section 2 details how the connection tracking process in typical stateful inspection firewalls interact with the log generation process, highlighting some particularities that arise from this interaction, especially in Linux's IPTables and Checkpoint's Firewall-1. Section 3 describes how the ideas of stateful connection tracking and correlation can be applied successfully in log analysis to expose a few kinds of security-related anomalies. Section 4 outlines some ideas for implementing these techniques directly on firewalls, intrusion detection systems and other network devices subject to real-time processing constraints. Section 5 presents conclusions and ideas for future work and implementations.

Este artigo está organizado como se segue: a seção 2 detalha como o processo de acompanhamento de conexões em *firewalls* com inspeção de estado típicas interage com o processo de geração do *log*, enfatizando algumas particularidades que advêm dessa interação, especialmente no IPTables do Linux e no Firewall-1 da Checkpoint. A seção 3 descreve como as idéias de acompanhamento e correlação baseada em estado podem ser aplicadas de forma bem sucedida

à análise de *logs*, de forma a expor alguns tipos de anomalias relacionadas à segurança. A seção 4 esboça algumas idéias para implementar essas técnicas diretamente nas *firewalls*, sistemas de detecção de intrusão e outros dispositivos de rede sujeitos a restrições de processamento em tempo real. A seção 5 apresenta as conclusões e idéias para trabalhos e implementações futuras.

2 TYPICAL STATEFUL LOG GENERATION
A TÍPICA GERAÇÃO DE *LOGS* COM ESTADO

Stateful inspection firewalls log their activities depending on the protocol involved: for TCP and UDP, they log the first packet that causes the "connection" to be evaluated against its rule base; subsequent packets of accepted connections are not logged since they are already in the state table. The return packets of these connections are not logged either, and are allowed to pass. Rejected connections are always logged since they necessarily cause the rule base to be consulted.

As *firewalls* com inspeção de estado registram suas atividades no *log* dependendo do protocolo envolvido: para TCP e UDP, elas registram o primeiro pacote que causa a "conexão" ser avaliada contra sua base de regras; pacotes subseqüentes de conexões aceitas não são registrados, uma vez que eles já se encontram na tabelas de estado. Os pacotes de retorno dessa conexão também não são registrados e sua passagem é permitida. Conexões rejeitadas são sempre registradas, dado que elas necessariamente causam a base de regras ser consultada.

A connection is identified by its traditional 4-tuple: (source address, source port, destination address, destination port). This allows the concept of "connection" to be extended to UDP, which, being a datagram protocol, lacks the concept of "connection"; in this paper, we refer to them as UDP "sessions".

Uma conexão é identificada pela sua 4-tupla tradicional: (endereço de origem, porta de origem, endereço de destino, porta de destino). Isso permite ao conceito de conexão ser estendido também para UDP, que, sendo um protocolo de datagrama, não tem o conceito de "conexão"; nesse artigo, referir-nos-emos a eles como "sessões" UDP.

The state tables have inactivity timeouts. Idle TCP connections are removed from the state table after a certain period (typically tens of minutes). The same applies to UDP sessions, but with a much shorter timeout – tens of seconds, typically. A connection that resumes transmitting after these periods is rechecked against its rule base and logged again.

As tabelas de estado têm expiração por inatividade. Conexões TCP ociosas são removidas da tabela de estado após um certo período (tipicamente [da ordem de] dezenas de minutos). O mesmo se aplica às sessões UDP, mas com um tempo de expiração muito mais curto – dezenas de segundos, tipicamente. Uma conexão que volta a transmitir após esse período é checada novamente contra a base de regras e registrada novamente.

For TCP, either a FIN or a RST packet removes the connection from the state tables. It is interesting to note that while connection initiations are logged, connection terminations are not. This is unfortunate, since interesting data could be obtained from them, such as the duration of the connection. This obviously doesn't apply to UDP, since it lacks explicit termination; they are dealt with by the aforementioned timeout rules.

Para TCP, tanto um pacote FIN quanto um RST remove a conexão das tabelas de estado. É interessante notar que, enquanto os inícios das conexões são registrados, os términos não são. Isso é infeliz, pois dados interessantes poderiam ser obtidos deles, tais como a duração da conexão. Isso obviamente não se aplica a UDP, uma vez que ele não tem término explícito; elas são tratadas pelas regras de expiração supracitadas.

ICMP packets, however, are typically logged in the traditional stateless fashion: each packet generates a log entry, without trying to prevent further log entries by relating it with previous exchanges. This is somewhat inconsistent, since there is usually enough information in ICMP packets to be able to relate requests and replies, despite the fact that they were not designed to provide connection services.

Pacotes ICMP, entretanto, são tipicamente registrados da maneira tradicional, sem estado: cada pacote gera uma entrada no *log*, sem nenhuma tentativa de evitar mais entradas no *log* através da correlação com intercâmbios anteriores. Isso é um tanto inconsistente, uma vez que usualmente há informação suficiente nos pacotes ICMP para se ser capaz de correlacionar pedidos e respostas, a despeito do fato de eles não terem sido projetados para prover serviços de conexão.

Checkpoint's Firewall-1 has an extra "excessive log" filtering feature built-in: nearly identical packets arriving within a certain time frame (62 seconds, by default) will not be logged. This prevents long-lasting ping trains, commonly used by system administrators and dynamic routing protocols for connectivity testing, to flood the logs with repetitive uninteresting data. IPTables can do that kind of rate limitation using token bucket filters, but it's not enabled by default.

O Firewall-1 da Checkpoint tem um recurso embutido extra de filtragem de "*log* excessivo": pacotes quase idênticos chegando dentro de um certo espaço de tempo (62 segundos, por *default*)

não são registrados. Isso evita que trens de *pings* de longa duração, comumente usados por administradores de sistemas e protocolos de roteamento dinâmicos para teste de conectividade, inundem os *logs* com dados repetitivos e desinteressantes. O IPTables também pode fazer esse tipo de limitação de taxa usando "token bucket filters", mas isso não é habilitado por padrão.

It is also appropriate to remind that logging is optional, being enabled or disabled on a rule-by-rule basis. The more pervasive the logging policy is, the better the results will tend to be.

É também apropriado relembrar que o ato de registrar no *log* é opcional, sendo ativado ou desativado regra a regra. Quanto mais abrangente a política de registro for, tanto melhor os resultados tenderão a ser.

In fact, Firewall-1 goes beyond, allowing each rule to be logged in one of several styles: "none" (no logging), "short" (logs only source and destination addresses and ports), "long" (same as "short" plus translated addresses and VPN events) and "accounting" (same as "long" plus total amount of data transferred). The more detailed the logging, the bigger the speed and space requirements.

De fato, o Firewall-1 vai além, permitido a cada regra ser registrada de várias formas: "nenhum" (nada é registrado), "short" (registra apenas endereços e portas de origem e destino), "long" (o mesmo que "short" mais endereços traduzidos e eventos da VPN) e "accounting" (o mesmo que "long" mais o total de dados transferidos). Quanto mais detalhado o *log*, tanto maior os requisitos de espaço e velocidade.

2.1 Related connection logging
Registrando conexões correlatas

When a connection is accepted by a rule with certain "special" destination ports (21/tcp, for example, corresponding to the control port of the FTP service), it is put "on watch": all packets in this connection are inspected looking for control information about new endpoint (addresses & ports) negotiations. In our example with the FTP protocol, that would be the "PORT" command. This new endpoint is added to a separate table, along with a destination endpoint. Together they become a special temporary rule that is checked even before the rule base and allows those connections to pass even without explicit mention in the rule base. This is what we call "related connections".

Quando uma conexão é aceita por uma regra com certas portas de destino "especiais" (21/tcp, por exemplo, correspondendo à porta de controle do serviço de FTP), ela é colocada "em observação": todos os pacotes dessa conexão são inspecionados à procura de informações de controle sobre negociação de novos *endpoints* (endereços e portas). No nosso exemplo com o protocolo FTP, esse seria o comando "PORT". Esse novo *endpoint* é adicionado a uma tabela separada, juntamente com o *endpoint* de destino. Juntas, elas se tornam uma regra temporária especial que é checada antes mesmo da base de regras e permite a essas conexões passarem mesmo sem menção explícita na base de regras. Estas são o que chamamos de "conexões correlatas".

When one of these related connections is initiated, the "related connections special rule table" is checked, a match is found and the connection is automatically accepted. However, Firewall-1 doesn't log the acceptance of these connections, maybe because they're supposed to be accepted anyway. This is unfortunate from the point of view of the log analyzer, since valuable information about the exact connections that took place is lost. IPTables, however, can be set up in a way that logs these connections.

Quando uma dessas conexões correlatas é iniciada, a "tabela especial de regras de conexões correlatas" é checada, um batimento é encontrado e a conexão é automaticamente aceita. Entretanto, o Firewall-1 não registra a aceitação dessas conexões, talvez porque elas supostamente são para serem aceitas mesmo. Isso é infeliz do ponto de vista do analisador de *logs*, uma vez que se perde informação valiosa sobre que conexões exatas tiveram lugar. IPTables, contudo, pode ser configurada de uma forma que registre essas conexões.

When the master connection is shut down, either by timeout or by explicit termination, all its related "special rules" are deleted, thus closing the "holes" they opened in the firewall. Notice that it doesn't finish any ongoing related connections; it merely prevents the creation of new ones. None of this is logged, though. It should also be stressed that the deleted rules are *temporary ones* kept in memory – none of this modifies the security policy rule table in any way.

Quando a conexão mestra é encerrada, seja por expiração ou por término explícito, todas as suas "regras especiais" correlatas são removidas, fechando assim os "buracos" que elas abriram na *firewall*. Note que isso não encerra nenhuma conexão correlata em andamento; isso meramente evita a criação de novas. Nada disso é registrado no *log*, entretanto. Deve-se também enfatizar que as regras removidas são *temporárias* mantidas em memória – nada disso modifica a base de regras da política de segurança de forma nenhuma.

While this process has been described for FTP only, it readily generalizes for several other protocols. The principle is the same: watch and interpret the control connection searching for new endpoint negotiations, adding them to the special

"related connections dynamic rule table" and making sure to get rid of them when the control connection finishes.

Apesar desse processo ter sido descrito apenas para FTP, ele prontamente pode ser generalizado para vários outros protocolos. O princípio é o mesmo: observe e interprete a conexão de controle à procura de novas negociações de *endpoints*, adicionando-s à "tabela especial de regras dinâmicas de conexões correlatas" e se certificado de se ver livre delas quando a conexão de controle terminar.

Note that this approach requires one handler for each application protocol, since it is necessary to understand the protocol messages in sufficient detail to extract the endpoint negotiations. Both Firewall-1 and IPTables have handlers for several popular protocols that require related connections, such as Sun RPC, RealAudio, etc.

Observe que essa abordagem requer uma rotina de tratamento para cada protocolo de aplicação, dado que é necessário entender as mensagens do protocolo em detalhe suficiente para extrair as negociações dos *endpoints*. Tanto o Firewall-1 quanto o IPTables têm módulos de tratamento para diversos protocolos populares que requerem conexões correlatas, tais como Sun RPC, RealAudio, etc.

It is unfortunate that neither Firewall-1 nor IPTables log the name of the application protocol handler or the endpoints of the related control connection – if that information was available, it would be possible to reconstruct exactly which related connection was generated by which control connection.

Infelizmente, nem o Firewall-1 nem o IPTables registram o nome da rotina de tratamento do protocolo de aplicação ou os *endpoints* da conexão de controle relacionada – se essa informação estivesse disponível, seria possível reconstruir exatamente que conexão correlata for gerada por qual conexão de controle.

From the preceding discussion, it follows that application protocols handled specially by the firewall will usually have only its control connection logged, preventing any correlation with its related connections – FTP, RealAudio, etc., being the prototypical examples. Several other protocols and network interactions, however, exhibit related connection behavior without being subject to any special processing. These are worthy cases for stateful correlation.

Da discussão acima, segue que os protocolos de aplicação tratados especialmente pela *firewall* geralmente só terão suas conexões de controle registradas, impedindo qualquer correlação com suas conexões correlatas – FTP, RealAudio, etc., sendo os exemplos prototípicos. Diversos outros protocolos e interações de rede, entretanto, exibem um comportamento de conexões correlatas sem serem sujeitos a nenhum processamento especial. Esses são casos dignos de nota para [a realização de] correlação com estado.

3 THE TECHNIQUE
A TÉCNICA

3.1 Connection correlation
Correlacionamento de Conexões

The following section of the log analyzer configuration file makes a good example of the technique:
O seguinte trecho do arquivo de configuração do analisador de *log* perfaz um bom exemplo da técnica:

```
1 port=80/tcp name=http-reverse-conn
2  master: record srcip, dstip
3  related-X: match srcip_r=dstip,
      dstip_r=srcip, dstport_r=6000/tcp
4  related-http: match srcip_r=dstip,
      dstip_r=scrip, dstport_r=80/tcp
5  related-ftp: match srcip_r=dstip,
      dstip_r=scrip, dstport_r=21/tcp
6  related-generic: match srcip_r=dstip,
      dstip_r=scrip, dstport_r=*/*
```

(The line numbers are for reference only in this text; they don't need to be present in the actual configuration file).

(Os números de linha são apenas para referência nesse texto; eles não precisam estar presentes no arquivo de configuração real).

The first line specifies the name of the event ("http-reverse-connection") and the port on which the master connections should be watched: 80/tcp. The second line specifies the which data from the master connection should be recorded in the "state table"; in this case, the source and destination IP addresses. We could simplify it by storing everything about the connection, but, since the state tables tend to grow quite large, it is more memory-efficient to store only what is effectively needed.

A primeira linha especifica o nome do evento ("http-reverse-connection") e a porta na qual a conexão mestra deve ser observada: 80/tcp. A Segunda linha especifica que dados da conexão mestra devem ser guardados na "tabela de estado"; nesse caso, os endereços IP de origem e destino. Poderíamos simplificar armazenando tudo sobre a conexão, mas, uma vez que as tabelas de estado tendem a crescer bastante, é mais eficiente em termos de memória armazenar somente o que é realmente necessário.

The remaining lines specify several cases of related connections that we would be interested to hear about:

As linhas restantes especificam os vários casos de conexões correlatas os quais estamos interessados em ouvir a respeito:

- The third line describes an attempt to connect to the X Windows port of the machine that originated the HTTP request. It often happens in Unix machines after a successful exploitation faulty CGI applications. Reading from the notation, it says “tag with the name ‘related-X’ all connections coming from the same IP of the destination of the master connection, going to the same IP that originated the master connection and whose destination port is 6000/tcp”.
- A terceira linha descreve uma tentativa de conectar na porta do X Windows da máquina que originou o pedido HTTP. Isso freqüentemente acontece em máquinas Unix após a exploração bem sucedida de aplicações CGI. Lendo a partir da notação, ele diz “rotule com o nome ‘related-X’ todas as conexões vindo do mesmo IP do destino da conexão mestra, indo para o mesmo IP que originou a conexão mestra e cuja porta de destino seja 6000/tcp”.
- The fourth and fifth directives go along similar lines, but for ports 80/tcp (HTTP) and 21/tcp (FTP). Readers with background on common exploits and intrusion detection will recognize this traffic behavior as arising from a successful exploitation of a common IIS vulnerability where the attacker connects elsewhere to download trojan horses or remote control programs.
- A quarta e quinta diretivas vão em linhas semelhantes, mas para as portas 80/tcp e 21/tcp. Leitores com familiaridade em *exploits* comuns e detecção de intrusão reconhecerão esse comportamento de tráfego como advindo da exploração bem sucedida de uma vulnerabilidade comum do IIS onde o atacante conecta em algum outro lugar para baixar cavalos de tróia ou programas de controle remoto.
- The sixth line is a “catch-all” for reverse connections: it would flag any connections originating from the web server originally contacted to the client that originally made the contact. The “*” stands for “any”.
- A sexta linha é um “pega-tudo” para conexões reversas: ela marcaria quaisquer conexões advindas do servidor web originalmente contatado para o cliente que originalmente fez o contato. O “*” representa “qualquer um”.

While none of these kind of traffic are proof of a security breach, they are uncommon enough to raise suspicions and deserve the attention of the administrators.

Ainda que nenhum desses tipos de tráfego seja prova [irrefutável] de um comprometimento da segurança, eles são incomuns o bastante para levantar suspeitas e merecer a atenção dos administradores.

It can be argued that the condition “dstip_r=srcip” is too restrictive – an attacker could download his backdoors from a machine other than the one he/she used to send the exploit. This condition could be relaxed if it is felt that it wouldn’t generate too many false alarms.

Pode-se argüir que a condição “dstip_r=srcip” é muito restritiva – um atacante poderia baixar seus *back doors* de uma máquina diferente da que ele/ela usou para enviar o *exploit*. Essa condição pode ser relaxada caso se sinta que isso não geraria muitos alarmes falsos.

On the other hand, if we make some assumptions about the security policy and the network architecture, we could generalize the match condition without significantly increasing its potential for false positives: if we assume that the HTTP servers are on a DMZ and the security policy forbids connections originating from the DMZ going to the Internet (a well-known Good Thing), we could write:

Por outro lado, se fizermos algumas suposições sobre a política de segurança e a arquitetura da rede, poderemos generalizar a condição de batimento sem significativamente aumentar o potencial para falsos positivos: se assumirmos que os servidores http estão em uma DMZ e que a política de segurança proíbe conexões originando-se a partir da DMZ indo para a Internet (uma bem conhecida Boa Coisa), poderíamos escrever:

```
6  related-generic: match srcip_r=dstip,
      dstport_r=*/*, action=reject or
      action=drop
```

That is, flag only the connections that were blocked – the fact that it was blocked is indication that it is against the security policy.

Isto é, marque apenas as conexões que tenham sido bloqueadas – o fato de ela ter sido bloqueada é indicação de que ela é contra a política de segurança.

This kind of correlation analysis is especially useful when doing forensic investigations in incident response scenario: it easily pinpoints reverse connections and other anomalies that are telltale signs of unauthorized activity, automating the tedious manual process of relevant evidence collection.

Esse tipo de análise de correlação é especialmente útil quando se está fazendo

investigações forenses em um cenário de resposta a incidentes: ela facilmente pinça as conexões reversas e outras anomalias que são a marca registrada de atividades não-autorizadas, automatizando o processo tedioso de coleta manual de evidências relevantes.

There are several kinds of traffic that can be correlated in this fashion. Although most of it is not directly security-related, the mere act of properly grouping them together and displaying it nicely encourage the system administrators to actually read the log summaries and thus conform to the classical "know thy traffic" security tenet. The following subsections illustrate some cases:

Há diversos tipos de tráfego que podem ser correlacionados deste modo. Apesar de muitos deles não serem diretamente relacionados à segurança, o mero ato de agrupá-los e mostrá-los apropriadamente encoraja os administradores de sistema a lerem de fato os sumários dos *logs* e assim se enquadrarem no clássico adágio de segurança "conheça teu tráfego". As subseções seguintes ilustram alguns casos:

3.2 ICMP Messages Correlation
Correlacionamento de mensagens ICMP

It would be useful to correlate the ICMP messages with the packets that originated them. The fragment below shows such a configuration in our tool for a simple UDP ⇔ ICMP correlation.

Seria útil correlacionar as mensagens ICMP com os pacotes que as originaram. O fragmento abaixo mostra a configuração da nossa ferramenta para um correlacionamento UDP ⇔ ICMP simples.

```
1 port=*/udp name=unreachables
2  master: record srcip, dstip,
                 ipid optional
3  port-unreach: match dstip_r=srcip,
      type=3-3/icmp, ipid_r=ipid
4  net-unreach: match dstip_r=srcip,
      type=3-0/icmp, ipid_r=ipid
5  host-unreach: match dstip_r=srcip,
      type=3-1/icmp, ipid_r=ipid
6  frag-needed: match dstip_r=srcip,
      type=3-4/icmp, ipid_r=ipid
7  admin-prohib: match dstip_r=srcip,
      type=3-13/icmp, ipid_r=ipid
   ...
```

The first line defines the "unreachables" tag and state table for all UDP sessions. The second line tells it to record only the source and destination IPs and the IP identification number. The following lines identify several kinds related ICMP control messages that could arise out of this packet: port, host or network unreachable, communication administratively prohibited (commonly sent by packet filtering routers), etc. In this example, the IP identification number is used to relate the replies with the packets that originated them. Since certain log file formats don't record the ID field of the IP header, the "optional" keyword is used to make the log analyzer try to relate the packets even in its absence. Without the "optional", the analyzer would simply discard this whole section due to lack of information to perform the correlation.

A primeira linha define o rótulo "unreachables" ["inalcançáveis"] e uma tabela de estado para todas as sessões UDP. A segunda linha diz para armazenar apenas os IPs de origem e destino e o número de identificação IP. As seguintes linhas identificam diversos tipos de mensagens que poderiam advir desse pacote: porta, *host* ou rede inalcançável, comunicação administrativamente proibida (comumente enviadas por roteadores com filtros de pacotes), etc. Nesse exemplo, o número de identificação IP é usado para correlacionar as respostas com os pacotes que as originaram. Uma vez que certos formatos de arquivos de *log* não registram o campo "ID" do cabeçalho IP, a palavra-chave "optional" é usada para fazer com que o analisador de *log* tente correlacionar os pacotes mesmo na sua ausência. Sem o "optional", o analisador simplesmente descartaria toda essa seção devido à falta de informação para realizar a correlação.

Even simple things such as correlating pings prove useful and have interesting security implications: (the example below was shortened for clarity – we could promptly add the same ICMP correlation rules we did above for UDP):

Até coisas simples como correlacionar pings se provam úteis e têm implicações de segurança interessantes: (o exemplo abaixo foi encurtado por claridade – poderíamos prontamente adicionar as mesmas regras de correlacionamento ICMP que fizemos acima para UDP):

```
1 type=8-0/icmp name=pings
2  echo-request: record srcip, dstip, ipid
3  echo-reply: match srcip_r=dstip,
      dstip_r=srcip, ipid_r=ipid optional,
      type=0-0/icmp, atmostonce
   ...
```

In the above example, the "atmostonce" keyword tells the analyzer that the replies must match the request at most once. "At most" because the reply may get lost or not be reported in the log. The tool considers an anomaly to see two or more replies to the same packet. Badly configured routing, broadcast address and other bizarre network effects might cause this and have been observed in practice. If the condition that requires the match of the IP IDs is relaxed, it might be used to detect ICMP tunnelers such as Loki – an interesting security event worth being flagged.

No exemplo acima, a palavra-chave "atmostonce" diz ao analisador que as respostas devem bater com os pedidos no máximo uma vez. "No máximo" porque a resposta pode ter sido perdida ou não ser reportada no *log*. A ferramenta considera uma anomalia ver duas ou mais respostas

a um mesmo pacote. Roteamento mal configurado, endereços de broadcast e outros efeitos bizarros de rede podem causar isso e foram observados na prática. Se a condição de batimento dos IP IDs for relaxada, pode ser usado para detectar tuneladores ICMP tais como o Loki – um evento de segurança interessante digno de ser exposto.

3.3 Connection/Session Counting and Graphing Contagem e Geração de Gráficos de Conexões e Sessões

Besides the "record" directive, the specification of the master connection allows for other kinds of processing. The example below implements a port scan detector:

Além da diretiva "record", a especificação da conexão mestra [no arquivo de configuração do analisador de *log*] permite outros tipos de processamento. O exemplo abaixo implementa um detector de varredura de portas:

```
1 type=*/tcp name=portscan-detector
2  histogrm: match port=*/tcp or port=*/udp
        count dstport group_by srcip
        graph if count > $threshold
```

This setting does the following: for each source IP, the analyzer builds a hash table that counts the number of different destination ports in the TCP connections and UDP sessions it originated. If the number of connections is greater then `$treshold` (a macro that we once set to expand to 60 and never more changed it), it produces a histogram graph of the distribution of the ports. The original idea was to make a real histogram graph to be saved as a GIF file to be viewed in a web page, but since one of the requisites of our first version was to be text-only, it produces a three-line report like the one shown in Figure 1.

Essa configuração faz o seguinte: para cada IP de origem, o analisador constrói uma tabela de *hash* que conta o número de diferentes portas de destino dentre as conexões TCP e sessões UDP por ele originados. Se o número de conexões [distintas] for maior que `$threshold` (uma macro que certa vez definimos para expandir para 60 e nunca mais mudamos), ele produz um gráfico de histograma da distribuição das portas. A idéia original era fazer um gráfico de histograma real para ser salvo como um arquivo GIF para ser visto em uma página web, mas uma vez que um dos requisitos da nossa primeira versão era ser somente-texto, ela produz um relatório de três linhas tal como o mostrado na figura 1.

The first line lists the total number of connection attempts that matched, the source address and the total number of unique ports.

A primeira linha lista o número total de tentativas de conexão que bateram, o endereço de origem e o número total de portas distintas.

The second line is the privileged port number space from 0 to 1023, each character representing 16 ports. The "-" characters means that this "slot" of 16 ports received no "hits" or connection attempts. Numbers and letters are hex digits representing the number of hits each slot had.

A segunda linha é o espaço de números de portas privilegiadas de 0 a 1023, com cada caractere representando 16 portas. O caractere "-" significa que essa "faixa" de 16 portas não recebeu nenhum 'hit' ou tentativa de conexão. Os números e as letras são dígitos hexadecimais representando o número de tentativas de conexão que cada faixa teve.

The third line is the full port number space from 0 to 65535, each character representing a slot of 1024 ports. Again, a "-" represents no hits in that slot. The numbers and letters, however, have a different meaning: they are the count of hits in the slot divided by 32, represented in Radix-32. In other words, "1" means anything from 1-32 hits, "2" means 33-64 hits, up to "W", meaning 993 to 1024 hits. This is a way to make a compact text-only histogram.

A terceira linha é o espaço de números de portas completo de 0 a 65535, cada caractere representando uma faixa de 1024 portas. Novamente, um "-" significa nenhum *hit* nessa faixa. Números e letras, entretanto, têm um significando diferente: eles são a contagem de *hits* na faixa divididos por 32, representados em base 32. Em outras palavras, "1" significa qualquer coisa entre 1 a 32 *hits*, "2" significa 33-64 *hits*, até "W", que significa 993 a 1024 *hits*. Essa é uma maneira de fazer um histograma compacto em modo texto.

This scan is an example picked from our real world logs. Experience has shown that this kind of scan is usually generated by the options of the NMAP tool in TCP `connect()` scan mode, plus some other "probing around" – that is, when we scan ourselves using NMAP, the shape of the histogram is quite similar.

Essa varredura é um exemplo retirado dos nossos *logs* do mundo real. A experiência demonstrou que esse tipo de varredura é usualmente gerada pelas opções [padrão] da ferramenta NMAP em modo de varredura TCP `connect()`, mais algumas outras "sondagens pelas redondezas" – ou seja, quando varremos nós mesmos usando o NMAP, o formato do histograma é bem similar.

The current version of the tool does not show

Figure 0

the exact targets of the port scans, although we can get that information from other subreports. We are currently working on making the syntax for specifying nested subgroupings generating their own counts, histogram graphs and subreports – and making them fully graphical. What becomes clear is the vast space for analysis criteria.

A versão atual da ferramenta não mostra os alvos exatos das varreduras de portas, muito embora possamos obter essa informação de outros sub-relatórios. Estamos atualmente trabalhando em fazer a sintaxe para se especificar sub-agrupamentos aninhados gerando suas próprias contagens, gráficos de histograma e sub-relatórios. O que se torna claro é um vasto espaço para critérios de análise.

A non-obvious characteristic of this port scan detection scheme is that it does not expect the scan to be in increasing port number order like many other port scan detector tools do. Modern port scan detectors randomize the order they try the ports, but since our technique counts the total number of distinct ports, it catches these cases perfectly well.

Uma característica não-óbvia deste esquema de detecção de varreduras de portas é que ele não espera que a varredura se dê em ordem crescente de número de porta tal como muitos outros detectores de varredura de portas esperam. As ferramentas modernas de varredura de portas embaralham a ordem com que elas tentam as portas, mas já que nossa técnica conta o número total de portas distintas, ela pega esses casos perfeitamente bem.

The somewhat arbitrary decision that a port scan is when we get connections to more than 60 distinct ports is certainly debatable, but perfectly configurable. While analyzing single-day log files, it has been found to be quite acceptable. It is planned that future versions of our tool will allow for complex expressions to calculate this threshold.

A decisão um tanto arbitrária de que uma varredura de portas é quando obtemos conexões em mais de 60 portas distintas certamente é questionável, mas [também é] perfeitamente configurável. Quando analisando arquivos de *log* de um dia só, revelou-se bastante aceitável. Planeja-se que versões futuras da nossa ferramenta permitirão expressões complexas para

It is interesting to apply the tool and these techniques for very large log files – actually, we are working a version in which the state tables are stored on disk as B-trees, so as to be able to analyze month-long logs and bigger. Our preliminary results show several unexpected features, like distributed slow port scans.

É interessante aplicar a ferramenta e essas técnicas a arquivos de *log* muito grandes - de fato, estamos trabalhando em uma versão na qual as tabelas de estado são armazenadas de disco como árvores-B, de forma a ser capaz de analisar *logs* do tamanho de um mês e maiores. Nossos resultados preliminares mostram diversas características inesperadas, tais como varreduras de portas distribuídas lentas.

4 REAL-TIME APPLICATIONS APLICAÇÕES EM TEMPO REAL

The ideas described above were used to implement a batch log analysis tool: the log files of a certain period, typically a whole day, were collected and a report was produced. While this makes for interesting reading, it's natural to think of the next steps:

As idéias descritas acima foram usadas para implementar uma ferramenta de análise de *log* em lote: os arquivos de *log* de um certo período foram coletados e um relatório foi produzido. Apesar de isso prover uma leitura interessante, é natural se pensar nos próximos passos:

- Firewall devices could already analyze and report their data in this "stateful/correlating" way. Actions could even be taken based on conclusions regarding the correlation analysis. The difficulty with this idea stems from the fact that the state tables take a lot of memory. Strict expiration and discard policies for the table entries should be applied to keep them within reasonable bounds. It could also be argued that the increased memory demand could make the firewall more vulnerable to resource exhaustion attacks. Performance might also become a problem in very fast networks and slow processors.

- Os dispositivos de *firewall* poderiam analisar e reportar seus dados já nessa forma de "correlação/com estado". Ações poderiam ser tomadas com base na análise de correlação. A dificuldade com essa idéia se deriva do fato que as tabelas de estado consomem muita memória. Regras estritas de expiração e descarte deveriam ser aplicadas às entradas da tabela para mantê-la dentro de limites [de tamanho] razoáveis. Pode-se também argüir que a maior exigência de memória poderia tornar o *firewall* mais vulnerável a ataques de exaustão de recursos. A performance também poderia se tornar um problema em redes muito rápidas e processadores lentos.

- Intrusion Detection Systems might be a better candidate for this kind of analysis. Some of them already perform a some kind of stateful analysis and correlation, but most of them usually limit themselves to analyze the packet contents in search of known common attack signatures.

- Sistemas de Detecção de Intrusão poderiam ser candidatos melhores para esse tipo de análise. Alguns deles já realizam algum tipo de análise e correlação com estado, mas a maioria deles geralmente se limita a analisar o conteúdo dos pacotes em busca de assinaturas conhecidas de ataques comuns.

- At the very least, firewalls should log more data, like connection termination; related connections caused by application-layer handlers; the exact identification of the application handler and the endpoint negotiation it detected; perhaps even the complete transport and network headers and the beginning of the payload – the goal being to make the log analyzer capable of accurately reconstruct the actions taken by the firewall and the interaction between the communicating parties.

- No mínimo, os *firewalls* deveriam registrar mais dados, tais como o término das conexões; conexões correlatas com as quais as rotinas de tratamento de protocolos de aplicações lidaram; a identificação exata dessas rotinas de tratamento e a negociação dos *endpoints* por elas detectados; talvez até os cabeçalhos completos dos [protocolos de] transporte e rede, mais o início da carga - o objetivo sendo o de capacitar o analisador de *log* a reconstruir precisamente as ações tomadas pelo *firewall* e a interação entre as partes em comunicação.

5 CONCLUSIONS AND FUTURE WORK CONCLUSÕES E TRABALHOS FUTUROS

This work showed that the same techniques used by stateful firewalls to filter the traffic can be applied to the field of log analysis. The characterization technique has been applied to expose several kinds of security-related incidents, such as reverse connections and covert channels. Many other types of anomalies, not necessarily security-related, can also be flagged.

Este trabalho mostrou que as mesmas técnicas usadas por *firewalls* com [inspeção de] estado podem ser aplicadas à área de análise de *logs*. A técnica de caracterização foi aplicada para expor diversos tipos de incidentes relacionados à segurança, tais como conexões reversas e canais disfarçados. Muitos outros tipos de anomalias, não necessariamente relacionadas à segurança, também podem ser evidenciadas.

Some inconsistencies and omissions in the way common stateful inspection firewalls generate their logs have been presented, especially regarding the stateless handling of ICMP packets, the omission of related application-level connections (FTP being the typical example). Most log files fail to provide enough information to accurately reconstruct their actions and some improvements were suggested.

Algumas inconsistências e omissões na maneira como os *firewalls* com inspeção de estado geram seus *logs* foram apresentadas, especialmente no que tange ao tratamento de pacotes ICMP, a omissão de conexões correlatas de nível de aplicação (FTP sendo o exemplo típico). A maioria dos arquivos de *log* falham em prover informações suficientes para se reconstruir precisamente suas ações e algumas melhorias foram sugeridas.

It has also been shown that state tables can be used to draw histograms or perform statistical characterization of the traffic that could be used to detect security-related probing, such as port scanning, or anomalous traffic patterns.

Também se mostrou que as tabelas de estado podem ser usadas para gerar histogramas e realizar uma caracterização do tráfego que pode ser usado para detectar sondagens relacionadas à segurança, tais como varreduras de portas, ou padrões de tráfego anômalos.

The tool implementing these techniques makes its analysis in batch mode, operating on a large text-mode log file. It was originally conceived both as a forensic analysis tool and a daily summarizer to be run along the log file rotation and archival process. However, it has been shown that the correlation techniques may also be implemented directly in the firewalls or in intrusion detection systems. A worthwhile goal in sight would be to produce patches to IPTables or Snort to achieve this.

A ferramenta que implementa essas técnicas realiza suas análises em lote, operando em um grande arquivo de *log* em modo texto. Ela foi originalmente concebida tanto como uma ferramenta de análise forense quanto um sumarizador diário para ser rodado juntamente com o processo de rotacionamento e arquivamento dos arquivos de *log*. Entretanto, mostrou-se que as técnicas de correlação também podem ser implementadas diretamente em *firewalls* ou sistemas de detecção de intrusão. Um objetivo meritório em vista seria produzir *patches* para o IPTables ou o Snort para realizar isto.

Another avenue of work being pursued is the statistical characterization of port scans and signature-based recognition of the tools that produced the scan – we would like our tools to be able to say something along the lines of “this anomaly is consistent with a NMAP `connect()` scan”.

Uma outra avenida de trabalho sendo perseguida é a caracterização estatística das varreduras de portas e o reconhecimento baseado em assinaturas das ferramentas que produziram a varredura – gostaríamos que nossas ferramentas fossem capazes

de dizer algo nas linhas de “essa anomalia é consistente com uma varredura NMAP `connect()`”.

6 REFERENCES
REFERÊNCIAS

STEVENS, W. Richard, *TCP/IP Illustrated, Volume 1 – The Protocols*, Addison-Wesley, 1994, ISBN 0-201-63346-9.

CHECKPOINT Software Technologies Ltd., *Checkpoint Firewall-1 Architecture and Administration*, 1998, Part No 71300001400.

iptables(8) man pages.

Snort IDS: `http://www.snort.org`

Project Loki: ICMP Tunneling: `http://www.phrack.org/show.php?p=49&a=6`

Phoneboy’s Firewall-1 FAQ: `http://www.phoneboy.com`

SPITZNER, Lance, *Understanding the FW-1 State Table: How Stateful is Stateful Inspection?*, http://www.enteract.com/~lspitz/fwtable.html

NORTHCUTT, Stephen, *Network Intrusion Detection: An Analyst's Handbook, 2nd Edition*, 2000, New Riders Publishing, ISBN 0735710082.

AMOROSO, Edward. *Intrusion Detection – An introduction to Internet surveillance, correlation, trace back, trap, and response*. AT&T Laboratories. Intrusion.Net Books, 1999, ISBN 0966670078.