Faculdade Senac Goiás
Curso de Gestão de Tecnologia da Informação
Professor Orientador: Tsukas
Acadêmico: Elson Cristino Farias
Acadêmico: Flamaryon Klever Santos Costa
Acadêmico: Lucas Eduardo Rodrigues Couto

Laboratório de Redes

Arquivos de logs Apache

Os logs são fontes riquíssimas de informações e são gerados pelos servidores e pelas aplicações conforme eventos significativos acontecem. Log é definido como um conjunto de registros com marcação temporal, que suporta apenas inserção, e que representa eventos que aconteceram em um computador ou equipamento de rede. Os registros de log constituem uma fonte básica de informação tanto para a detecção, como para a resolução de problemas. Com a análise destas informações providas pelos logs é possível detectar o uso indevido do ambiente de TI, ataques, exploração de vulnerabilidades, rastrear (auditar) as ações executadas por um usuário e detectar problemas de hardware ou nos programas e serviços instalados no computador.

Uma auditoria de segurança bem-sucedida depende da existência de registros de logs íntegros e confiáveis. Independente do quão seguro é um computador, uma rede ou um sistema, nunca será possível confiar totalmente nos registros de um sistema que foi comprometido, pois isso dificulta ou até mesmo impossibilita uma auditoria de sucesso. Quando os registros de auditoria estão seguros é possível aumentar as chances de sucesso ao se correlacionar e identificar padrões ou rever os incidentes de segurança ocorridos em um sistema. Para alcançar estes objetivos é recomendável estabelecer um sistema de logs centralizado e dedicado, ou seja, que tenha como função exclusiva a coleta, registro e análise de eventos de logs.

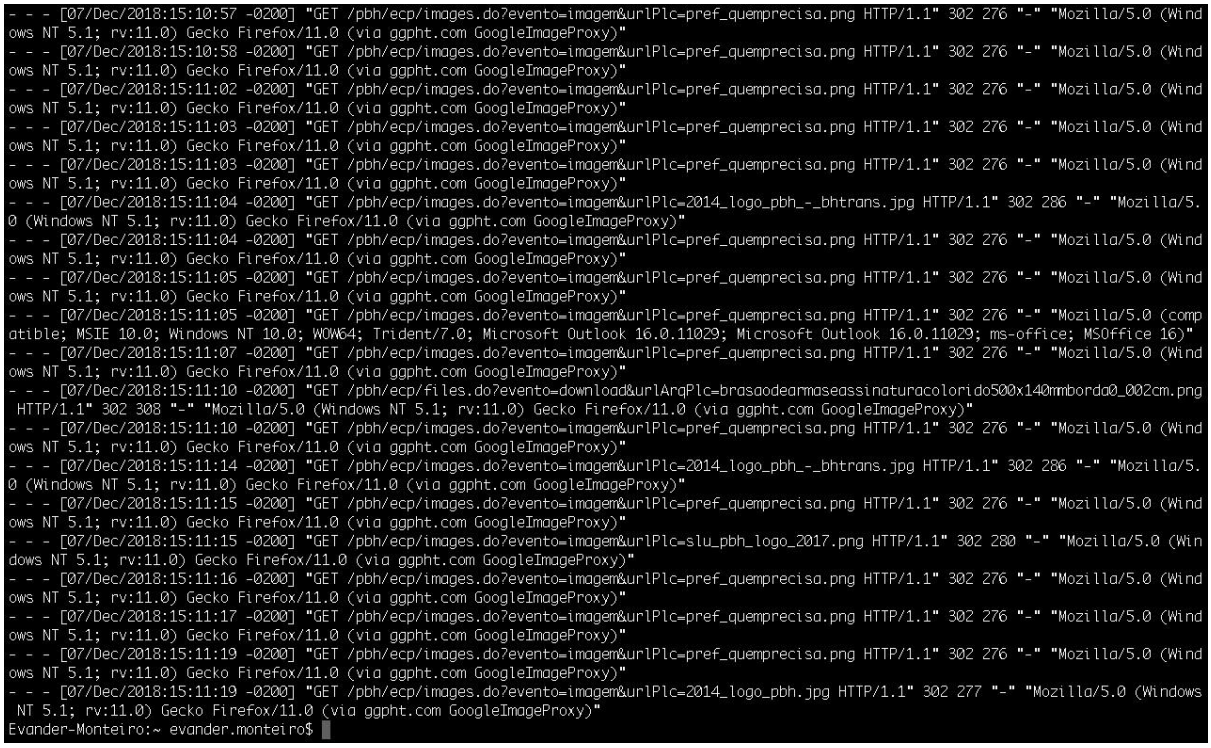
Com base nestas informações é possível tomar medidas preventivas para tentar evitar que um problema maior ocorra ou, caso não seja possível, tentar reduzir os danos. O conjunto de atividades desempenhadas na operação de logs, como visualização, armazenamento, compactação, e assim por diante, pode ser denominado de processo de gerenciamento de logs. A importância deste processo é evidenciada por publicações que orientam como as atividades deste processo devem ser conduzidas.

Para gerenciar com eficiência um servidor da Web, é necessário obter feedback sobre a atividade e o desempenho do servidor, bem como sobre quaisquer problemas que possam estar ocorrendo. O Apache HTTP Server fornece recursos de registro muito abrangentes e flexíveis, possibilitando utilizar um arquivo de log único ou diversos arquivos de logs registrando cada evento ocorrido na aplicação, informações de conexão, navegador, bloqueio de acesso, erros e outros.

O Apache HTTP Server fornece uma variedade de mecanismos diferentes para registrar tudo o que acontece em seu servidor, desde a solicitação inicial, passando pelo processo de mapeamento de URL, até a resolução final da conexão, incluindo quaisquer erros que possam ter ocorrido no processo. Além disso, os módulos de terceiros podem fornecer recursos de registro ou inserir entradas nos arquivos de log existentes, e aplicativos como programas CGI, scripts PHP ou outros manipuladores podem enviar mensagens para o log de erros do servidor.

Qualquer pessoa que possa escrever para o diretório em que o Apache httpd está gravando um arquivo de log quase certamente obterá acesso ao uid iniciado pelo servidor, que normalmente é o root. Você não deve dar acesso de gravação para qualquer pessoa no diretório onde os logs são armazenados sem estar consciente das consequências.

O Apache é bem flexível na especificação do que será registrado em seus arquivos de log, possibilitando utilizar um arquivo de log único, diversos arquivos de logs registrando cada evento ocorrido no sistema (conexão, navegador, bloqueio de acesso, erros, etc) incluindo os campos que deseja em cada arquivo e a ordem dos campos em cada um deles. Enfim qualquer coisa pode ser especificada de forma que atenda as suas necessidades particulares de logging.



É possível usar processamento de log para extrair diversas informações. Um de seus usos mais comuns é extrair erros ou contar a ocorrência de algum evento dentro de um sistema (como falhas de logins). Também é possível extrair alguns tipos de dados de desempenho, como conexões ou transações por segundo. Outras informações úteis incluem a extração (mapa) e criação de vistas de sites (redução) de um log da web. Essa análise também pode suportar a detecção de visitas de usuários exclusivas, além de estatísticas de acesso a arquivo.

Basicamente o Apache utiliza dois Logs centralizadores, Error_log e Access_log sendo Error_log para gravar mensagens de erro e alertas e access_log para gravar informações de conexão. É possível criar um arquivo de log para cada site, basta configurar no virtualhost.

Log de Erros (Error_log)

O log de erros do servidor, cujo nome e local é definido pela diretiva ErrorLog, é o arquivo de log mais importante. Este é o local em que o Apache httpd enviará informações de diagnóstico e registrará quaisquer erros encontrados em solicitações de processamento. É o primeiro lugar para procurar quando ocorre um problema com no servidor ou com uma operação do servidor, uma vez que ele geralmente contém detalhes do que deu errado e como consertálo. Deve habilitar a gravação do error_log no arquivo de configuração do apache, em um CentOS, em /etc/httpd/conf/httpd.conf seguindo o modelo abaixo.

```
#
ErrorLog logs/error_log
LogLevel warn
```

O log de erros geralmente é gravado em um arquivo (normalmente error_log no arquivo error.log). Em sistemas Unix, também é possível fazer com que o servidor envie erros syslog.

O formato do log de erros é definido pela diretiva ErrorLogFormat, com a qual você pode personalizar quais valores são registrados. Um padrão é definido se você não especificar um.

Abaixo uma mensagem típica de log:

```
Fri Dec 07 09:39:55 2018] [error] [client 66.249.83.41] File does not exist: /var/www/html/consulta/public_html/favicon.ico
```

O primeiro item na entrada de log é a data e hora da mensagem, em seguida o nível de severidade dessa mensagem, em seguida, temos o endereço do cliente que fez a solicitação. E finalmente é a mensagem de erro detalhada, que neste caso indica uma solicitação para um arquivo que não existia.

Uma grande variedade de mensagens diferentes pode aparecer no log de erros. A maioria parece semelhante ao exemplo acima. O log de erros também conterá a saída de depuração de scripts CGI. Qualquer informação gravada stderrpor em um script CGI será copiado diretamente para o log de erros.

Colocar um %Ltoken no log de erros e no log de acesso produzirá um ID de entrada de log com o qual você pode correlacionar a entrada no log de erros com a entrada no log de acesso. Se mod_unique_idfor carregado, seu ID de solicitação exclusivo será usado como o ID de entrada de log também.

Em sistemas Unix é possível filtrar os logs com o comando egrep ou adicionar | grep ao final do comando de leitura, sendo o comando cat (nós utilizamos) ou o comando tail que pega as ultimas linhas do arquivo, já que o parâmetro -f faz com que o arquivo fique constantemente sendo lido para que seja possível acompanhar em tempo de execução um passo a passo.

```
[root@ganweb httpd]# tail -f error_log
```

```
[root@ganweb ~]# cat /var/log/httpd/error_log | grep favicon
```

Log de Acessos (access_log)

O log de acesso do servidor registra todas as solicitações processadas pelo servidor. A localização e o conteúdo do log de acesso são controlados pela diretiva CustomLog. A diretiva LogFormat pode ser usada para simplificar a seleção do conteúdo dos logs. Obviamente, armazenar as informações no log de acesso é apenas o início do gerenciamento de logs.

O próximo passo é analisar essas informações para produzir estatísticas úteis. A análise de logs em geral está além do escopo deste documento e não faz parte do trabalho do próprio servidor da web. Para obter mais informações sobre esse tópico e para aplicativos que executam a análise de log.

Vamos dar um exemplo que usa o formato comum de log Apache:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

Cada seção desse log significa:

%h - O endereço IP do cliente.
%l- A identidade do cliente determinada identdna máquina do cliente. Retornará um hífen () se esta informação não estiver disponível.
%u - O ID do usuário do cliente, se a solicitação foi autenticada.
%t - A hora em que a solicitação foi recebida.
\"%r\"- A linha de solicitação que inclui o método HTTP usado, o caminho do recurso solicitado e o protocolo HTTP usado pelo cliente.
%>s - O código de status que o servidor envia de volta ao cliente.
%b - O tamanho do objeto solicitado.

Se uma solicitação foi feita para um site usando o formato de log acima mencionado, o log resultante será semelhante ao seguinte.

```
201.78.143.67 – DestopNh21 [19 / Mar / 2018: 10: 34: 12 -0700] "GET / sample-imagem.png HTTP / 2" 200 1479
```

Caso precise de um pouco mais de granularidade com seus logs de acesso, poderá usar o formato de log personalizado do Apache. Usando o módulo de log personalizado, você precisa definir em seu arquivo de configuração do Apache onde deseja armazenar o log e o formato que deseja usar.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

Nesse formato registra logs com informações de erro 400, 501, 200, 304 e 302. Os modificadores "<" e ">" podem ser usados para solicitações que foram redirecionadas internamente para escolher se a solicitação original ou final (respectivamente) deve ser consultada. Por padrão, as % diretivas %s, %U, %T, %D e % reexaminam a solicitação original, enquanto todas as outras examinam a solicitação final. Por exemplo, %>s pode ser usado para registrar o status final da solicitação e %<u pode ser usado para registrar o usuário autenticado original em uma solicitação redirecionada internamente para um recurso não autenticado.

Pode-se utilizar os seguintes formatos de log personalizado:

Cadeia de formato	**Descrição**
`%%`	O sinal de porcentagem.
`%a`	Endereço IP do cliente da solicitação (consulte o `mod_remoteip` módulo).
`%{c}a`	Endereço IP de mesmo nível subjacente da conexão (consulte o `mod_remoteip` módulo).
`%A`	Endereço IP local.

`%B`	Tamanho da resposta em bytes, excluindo cabeçalhos HTTP.
`%b`	Tamanho da resposta em bytes, excluindo cabeçalhos HTTP. No formato CLF, *ou seja* , um ' −' em vez de um 0 quando nenhum byte é enviado.
`%{`*VARNAME*`}C`	O conteúdo do cookie *VARNAME* na solicitação enviada ao servidor. Somente os cookies da versão 0 são totalmente suportados.
`%D`	O tempo gasto para atender à solicitação, em microssegundos.
`%{`*VARNAME*`}e`	O conteúdo da variável de ambiente *VARNAME* .
`%f`	Nome do arquivo.

`%h`	Nome do host remoto. Registrará o endereço IP se `HostnameLookups` estiver definido como `Off`, que é o padrão. Se ele registrar o nome do host apenas para alguns hosts, você provavelmente terá diretivas de controle de acesso mencionando-as pelo nome. Consulte a documentação do host requer .
`%H`	O protocolo de solicitação.
`%{`*VARNAME*`}i`	O conteúdo das *VARNAME*: linhas de cabeçalho na solicitação enviada ao servidor. Alterações feitas por outros módulos (por exemplo `mod_headers`) afetam isso. Se você estiver interessado em saber qual era o cabeçalho da solicitação antes de a maioria dos módulos modificá-lo, use `mod_setenvif` para copiar o cabeçalho em uma variável de ambiente interna e registrar esse valor com o descrito acima. `%{`*VARNAME*`}e`

`%k`	Número de pedidos de keepalive manipulados nesta conexão. Interessante se `KeepAlive` estiver sendo usado, de modo que, por exemplo, um '1' significique o primeiro pedido de manutenção de atividade após o inicial, '2' o segundo, etc ...; caso contrário, isso é sempre 0 (indicando a solicitação inicial).
`%l`	Logname remoto (do identd, se fornecido). Isso retornará um traço, a menos que `mod_ident` esteja presente e `IdentityCheck` esteja definido `On`.
`%L`	O ID do log de solicitação do log de erros (ou '-' se nada tiver sido registrado no log de erros para esta solicitação). Procure a linha do log de erros correspondente para ver qual solicitação causou o erro.
`%m`	O método de solicitação.
`%{`*VARNAME*`}n`	O conteúdo da nota *VARNAME* de outro módulo.
`%{`*VARNAME*`}o`	O conteúdo da *VARNAME*: (s) linha (s) de cabeçalho na resposta.
`%p`	A porta canônica do servidor que atende a solicitação.
`%{`*format*`}p`	A porta canônica do servidor que atende a solicitação, ou a porta real do servidor ou a porta real do cliente. Formatos válidos são `canonical`, `local` ou `remote`.

`%P`	O ID do processo da criança que atendeu a solicitação.
`%{`*format*`}P`	O ID do processo ou o ID do segmento do filho que atendeu a solicitação. Formatos válidos são `pid`, `tid` e `hextid`. `Hextid` requer APR 1.2.0 ou superior.
`%q`	A string de consulta (prefixada com uma string de consulta existir, caso contrário, uma string vazia).

`%r`	Primeira linha de solicitação.
`%R`	O manipulador que gera a resposta (se houver).
`%s`	Status. Para solicitações que foram redirecionadas internamente, esse é o status da solicitação *original*. Use `%>s` para o status final.
`%t`	Horário em que a solicitação foi recebida, no formato `[18/Sep/2011:19:18:28 -0400]`. O último número indica o deslocamento do fuso horário do GMT

`%{`*format*`}t`	A hora, na forma dada pelo formato, que deve estar em um `strftime`formato estendido (potencialmente localizado). Se o formato começar com `begin:`(padrão), a hora será tomada no início do processamento da solicitação. Se começar `end:`, é a hora em que a entrada de log é gravada, perto do final do processamento da solicitação. Além dos formatos suportados `strftime`, os seguintes formatos de tokens são suportados:
	
	`sec` número de segundos desde a época `msec`
	número de milissegundos desde a época `usec`
	número de microssegundos desde a época
	`msec_frac` fração de milissegundos `usec_frac`
	fração de microssegundos

	Esses tokens não podem ser combinados entre si ou `strftime` formatados no mesmo formato de string. Você pode usar vários tokens `%{`*format*`}t`

%T	O tempo gasto para atender à solicitação, em segundos.
%{*UNIT*}T	O tempo gasto para atender a solicitação, em uma unidade de tempo fornecida por UNIT. As unidades válidas são ms por milissegundos,us por microssegundos e s por segundos. Usar s fornece o mesmo resultado que %T sem qualquer formato; usando us dá o mesmo resultado que %D. Combinando %T com uma unidade está disponível em 2.4.13 e posterior.
%u	Usuário remoto se a solicitação foi autenticada. Pode ser falso se o status de retorno (%s) for 401 (não autorizado).
%U	O caminho da URL solicitado, não incluindo nenhuma string de consulta.
%v	O canônico ServerName do servidor que atende a solicitação.
%V	O nome do servidor de acordo com a UseCanonicalName configuração.
%X	Status da conexão quando a resposta é concluída: X = Conexão anulada antes de a resposta ser concluída. + = A conexão pode ser mantida ativa após a resposta ser enviada. - = A conexão será fechada após a resposta ser enviada.
%I	Bytes recebidos, incluindo solicitação e cabeçalhos. Não pode ser zero. Você precisa habilitar mod_logio para usar isso.
%O	Bytes enviados, incluindo cabeçalhos. Pode ser zero em casos raros, como quando uma solicitação é interrompida antes que uma resposta seja enviada. Você precisa habilitar mod_logio para usar isso.

%S	Bytes transferidos (recebidos e enviados), incluindo solicitação e cabeçalhos, não podem ser zero. Esta é a combinação de% I e% O. Você precisa habilitar `mod_logio` para usar isso.
%{*VARNAME*}^ti	O conteúdo da *VARNAME*:(s) linha (s) do trailer na solicitação enviada ao servidor.
%{*VARNAME*}^to	O conteúdo da *VARNAME*:(s) linha (s) de reboque na resposta enviada do servidor.

Tratamento de dados

Utilizamos o arquivo de log para tratamento das informações gravadas disponibilizado pelo link https://portalpbh.pbh.gov.br/logs/access.log, abaixo alguns prints de filtros de informações que deverão ser extraídas para análise.

1- Filtramos todos as solicitações do domínio google.com.br:

2- Todos acessos realizados por dispositivos Android:

3- Solicitações HTTP GET dos arquivos favicon.ico:

Conclusão

Os logs de acesso do Apache podem oferecer uma grande quantidade de informações sobre as solicitações recebidas para o seu servidor web. Se você precisar analisar esses logs em grandes quantidades, pode ser benéfico usar uma ferramenta de análise de logs que possa “processar os números” para você com muito mais rapidez.

A intenção desse documento foi demonstrar a importância da análise dos logs do Apache, por eles o profissional de TI poderá encontrar erros, ameaças e analisar o bom funcionamento do serviço.

Referências

Disponível em: http://www.ezequieljuliano.com.br/?p=76
Disponível em: https://www.maxwell.vrac.puc-rio.br/12571/12571_3.PDF
Disponível em: https://httpd.apache.org/docs/2.4/logs.html
Disponível em: https://portalpbh.pbh.gov.br/logs/access.log
Disponível em: http://httpd.apache.org/docs/current/mod/mod_log_config.html#formats
Disponível em: https://www.ibm.com/developerworks/br/library/os-log-processhadoop/index.html