Trabalho de conclusão do curso de Pós-graduação em Segurança da Informação pela Faculdade de Tecnologia IBRATEC de João Pessoa-PB em Jul/2013.

APLICANDO RECOMENDAÇÕES DE SEGURANÇA NA INSTALAÇÃO DO ZABBIX

Aécio dos Santos Pires
(aeciopires@gmail.com)

Resumo
O Zabbix é um sistema web que monitora vários parâmetros de uma rede, o status e a integridade dos serviços e equipamentos. Por ser uma aplicação web, o Zabbix herda as vulnerabilidades que qualquer aplicação web está sujeita. O objetivo deste trabalho é apresentar a instalação do Zabbix e dos serviços adjacentes aplicando as recomendações de segurança divulgadas em fóruns, palestras, manuais e tutoriais. Ao longo do artigo também são apresentadas as principais características do Zabbix e dos serviços adjacentes.

Palavras-chave: Instalação. Segurança. Zabbix.

Introdução

Zabbix é um *software* que monitora a disponibilidade e desempenho de aplicações de rede, parâmetros de uma rede, saúde e integridade de qualquer equipamento que possa ser acessado remotamente por um endereço IP (*Internet Protocol*). Com ele é possível agir proativamente, detectando e corrigindo falhas antes que os usuários percebam, além de analisar a disponibilidade dos serviços oferecidos na rede.

Por ser uma aplicação web, o Zabbix herda as vulnerabilidades que qualquer aplicação Web está sujeita, e pode sofrer vários tipos de ataques, tais como: SQL injection, Cross Site Script, força bruta, roubo de sessões HTTP (*Hypertext Transfer Protocol*), *path disclosure*, *defacement*, entre outros. Por isso é importante procurar por pontos vulneráveis no código fonte e, quando possível, aplicar medidas de prevenção no desenvolvimento do Zabbix, como também aplicar as recomendações de segurança na instalação e configuração do Zabbix e dos softwares adjacentes.

O objetivo deste artigo é apresentar as recomendações de segurança divulgadas em fóruns, palestras, manuais e tutoriais e aplicar a instalação do Zabbix e serviços adjascentes para que os riscos de ataques sejam minimizados. Neste artigo não serão apresentadas as configurações de segurança a serem aplicadas no sistema operacional porque variam muito e isto foge do escopo deste trabalho.

Para a elaboração deste artigo foi realizada a leitura de manuais escritos pelos desenvolvedores dos principais serviços adjacentes e do Zabbix, bem como houve a participação em cursos e palestras, que trataram desta temática, e várias interações com usuários mais experientes através dos fóruns e lista de discussão.

No ambiente de teste montado para a elaboração deste artigo foi utilizada apenas uma máquina virtual com as seguintes configurações:

- Processador: Intel(R) Core(TM) i3-2310M 2.10 GHz;
- Memória RAM: 2 GB;
- Disco rígido: 20 GB;
- Sistema operacional: Ubuntu Server 12.04 64 bits.

1 Serviços adjacentes

Para o Zabbix funcionar, é necessário o uso de um servidor Web (Apache), um sistema de gerenciamento de banco de dados e o interpretador PHP, linguagem de programação usada no desenvolvimento da interface Web do Zabbix.

O Zabbix tem suporte aos bancos de dados PostgreSQL, MySQL, Oracle, IBM DB2 e SQLite, mas neste artigo será usado o PostgreSQL, escolhido pela familiaridade que o autor adquiriu no uso diário.

É interessante falar sobre os serviços adjacentes ao Zabbix porque eles são importantes ao funcionamento do ambiente de monitoramento e também porque neles serão aplicadas boas práticas de configuração de segurança com objetivo de minimizar os riscos de ataque a aplicação Web.

1.1 Apache

"O Apache é um serviço flexível que implementa os mais recentes protocolos Web, incluindo HTTP/1.1. Ele é altamente configurável e extensível com módulos de terceiros. Funciona nos sistemas operacionais Windows, Netware 5.x ou superior, OS/2, e na maioria das versões do Unix" (WIKI APACHE). O protocolo HTTP/1.1 é especificado na RFC 2616 disponível em: http://www.ietf.org/rfc/rfc2616.txt.

O Apache está na versão 2.4.4, sendo mantido pela Apache Software Foundation e uma comunidade de desenvolvedores voluntários espalhados ao redor do mundo. Ele funciona tendo como base para o relacionamento entre cliente e servidor, no qual o cliente é um navegador, que tem como característica principal permitir a visualização de várias mídias, além do padrão básico HTML (*HyperText Markup Language*).

O servidor Web é acessado através do protocolo HTTP, que permite o tráfego das informações desejadas através de uma conexão TCP (*Transmission Control Protocol*). "A ideia do HTTP é muito simples. Um cliente envia um pedido, na forma de uma mensagem ao servidor. O servidor envia a resposta, também na forma de uma mensagem, ao cliente. As mensagens de pedido e resposta transportam dados na forma de documento. Os comandos do cliente ao servidor são inseridos numa mensagem de solicitação." (FOROUZAN, 2006, p. 641).

1.2 PHP

"PHP (*PHP Hipertext Preprocessor*) é uma linguagem de programação de código fonte aberto utilizada amplamente para fins gerais de uma linguagem de script. Foi originalmente projetada para ser usada no desenvolvimento de sites. Foi criada por Rasmus Lerdorf para auxiliar os usuários com tarefas relacionadas a Home Pages Web." (VALADE, 2004, p. 26).

Atualmente PHP encontra-se na versão 5.5.0 e foi usada no Zabbix para o desenvolvimento da interface Web.

1.3 PostgreSQL

"PostgreSQL é um servidor de banco de dados SQL avançado, disponível em uma ampla gama de plataformas. Um dos mais claros benefícios do PostgreSQL é o fato dele possuir o código aberto e uma licença muito permissiva para instalá-lo, usá-lo e distribuí-lo sem pagar qualquer taxa ou royalties" (RIGGS, 2010).

O PostgreSQL está na versão 9.2.4. O banco de dados é um componente importante ao Zabbix porque nele são armazenados os dados coletados dos equipamentos que estão sendo monitorados. Esses dados serão usados em relatórios alertas, senhas de usuários, templates de monitoramento e muito mais informações. Este componente pode estar instalado no mesmo computador do componente server ou não.

2 ZABBIX

Zabbix é um *software* que monitora a disponibilidade e desempenho de aplicações de rede, parâmetros de uma rede, a saúde e integridade de qualquer equipamento que possa ser acessado remotamente por um endereço IP.

Possui um mecanismo de notificação bastante flexível que permite avisar a ocorrência de eventos por e-mail, SMS, Jabber e, se for integrado a um shell script, pode avisar por Gtalk e Skype. Isto possibilita uma reação rápida aos problemas detectados. O Zabbix ainda oferece excelentes relatórios e visualização de dados armazenados em um banco de dados, tornando-o ideal para gerenciamento e planejamento da rede.

Todos os relatórios e estatísticas, bem como os parâmetros de configuração, são acessados através de uma interface Web desenvolvida em PHP (ver seção 1.2). Essa interface Web garante que as informações de cada equipamento que estiver sendo monitorado possam ser visualizadas e analisadas a partir de qualquer equipamento que possua um navegador Web e conectividade com o servidor Zabbix.

O Zabbix é um *software* totalmente de código fonte aberto, licenciado sob a licença GPL versão 2, e livre de custos. Ele foi criado por Alexei Vladishev e atualmente é desenvolvido e suportado comercialmente pela Zabbix SIA.

O Zabbix possui várias comunidades de usuários espalhadas ao redor do mundo que oferecem suporte extraoficial de forma voluntária e gratuita. No Brasil, existe a comunidade Zabbix-BR (zabbixbrasil.org), criada em 2008 por André Déo contando com a presença de mais 700 pessoas, que além de fornecer o suporte aos usuários, desenvolve pesquisas, difunde o uso do Zabbix em eventos tecnológicos e cria documentação na língua portuguesa.

2.1 Características

O Zabbix apresenta as seguintes características:

- Possui suporte a maioria dos sistemas operacionais: Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, NetBSD, MacOS X, Windows, entre outros;
- Monitora serviços simples (HTTP, POP3, IMAP, SSH) sem o uso de agentes;
- Suporte nativo ao protocolo SNMP (*Simple Network Management Protocol*);
- Interface de gerenciamento Web;
- Integração com banco de dados (MySQL, Oracle, PostgreSQL, IBM DB2 ou SQLite);
- Geração de gráficos em tempo real;
- Agentes disponíveis para diversas plataformas: Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, SCO-OpenServer, MacOS X, Windows Server 2000/XP/2003/Vista/7/8/2008;
- Agentes para plataformas 32 bits e 64 bits;
- Envio nativo de alertas para e-mail, Jabber e SMS;
- Integração com scripts personalizados;
- Uso e edição de templates com diversos parâmetros pré-definidos a serem usados para monitorar vários equipamentos com características similares;
- Monitoramento distribuído;
- Monitoramento de SLA;
- Monitoramento de aplicações JMX (*Java Management Extensions*), entre outras.

2.2 Componentes

O Zabbix é composto de vários componentes de *software*, os principais são:

- Zabbix Server: é o componente central do sistema. Ele pode verificar remotamente os serviços de rede (como serviço web e e-mail), utilizando a checagem simples, mas também é o componente central para que os agentes enviem informações e estatísticas acerca da disponibilidade e integridade do equipamento que está sendo monitorado. Depois que o servidor recebe essas informações, ele processa, gerencia os equipamentos, exibe relatórios, envia alertas e toma ações dependendo da configuração.
- Banco de dados: é o onde os dados, as informações e configurações são armazenadas. O banco de dados pode ser acessado diretamente pelo Zabbix server e pela interface Web. O Zabbix tem suporte aos bancos de dados: PostgreSQL, MySQL, Oracle, SQLite e IBM DB2.
- Interface web: é por ela que o Zabbix pode ser configurado e as informações visualizadas.
- Agente Zabbix: aplicação cliente do Zabbix que coleta informações do equipamento e envia ao servidor. O agente é capaz de acompanhar ativamente o uso dos recursos e aplicações locais, tais como: discos rígidos, memória, processador, processos, serviços e aplicativos em execução.
- Zabbix Proxy: é uma parte opcional do Zabbix. O Proxy coleta dados de desempenho e disponibilidade em nome de um servidor Zabbix, com a vantagem de coletar milhares de informações por segundo utilizando um hardware modesto. Podemos considerá-lo como um super agente.
- Java Gateway: O Zabbix 2.0 trouxe o suporte nativo ao monitoramento de aplicações JMX através do daemon Zabbix Java gateway. Quando o servidor Zabbix quer saber o valor de um contador de JMX de um host, ele pede a esse daemon, que usa a API (Application Programming Interface) de gerenciamento JMX para consultar a aplicação remotamente.

2.3 Funcionamento do Zabbix

O servidor Zabbix é formado pelo banco de dados, o componente server e a interface Web. Estas três partes podem ser instaladas no mesmo computador ou em computadores diferentes, conforme mostrado na Figura 1.

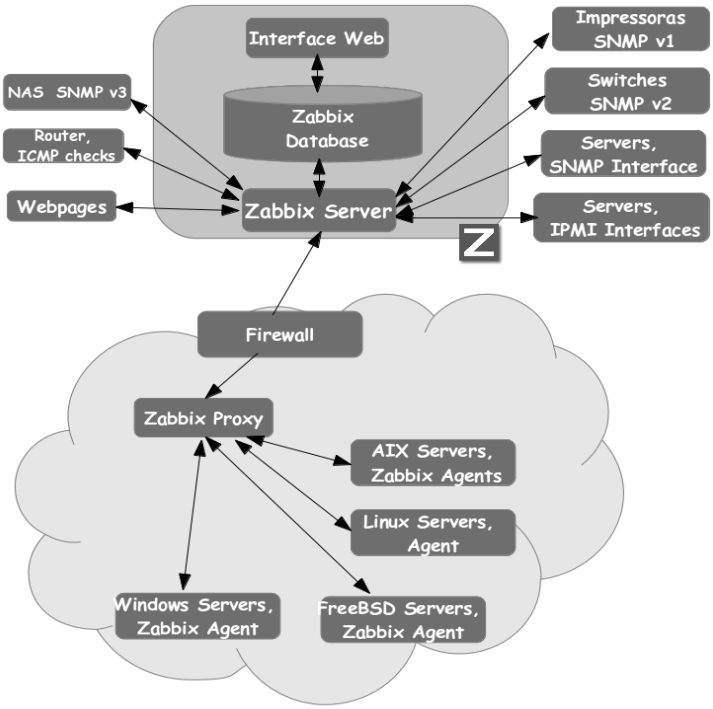


Figura 1 - Os componentes do Zabbix

Fonte: Adaptado de OLUPS, 2010, página 9.

A Figura 1 mostra que o componente server e a interface Web acessam frequentemente o banco de dados para obter ou armazenar informações. Essa figura também mostra que o servidor Zabbix pode receber os dados diretamente de diversos agentes ou pode receber os dados por meio do componente proxy, que por sua vez está recebendo os dados diretamente dos agentes e enviando-os de forma unificada para o componente server.

A Figura 2 mostra a interação entre os componentes agente e server.

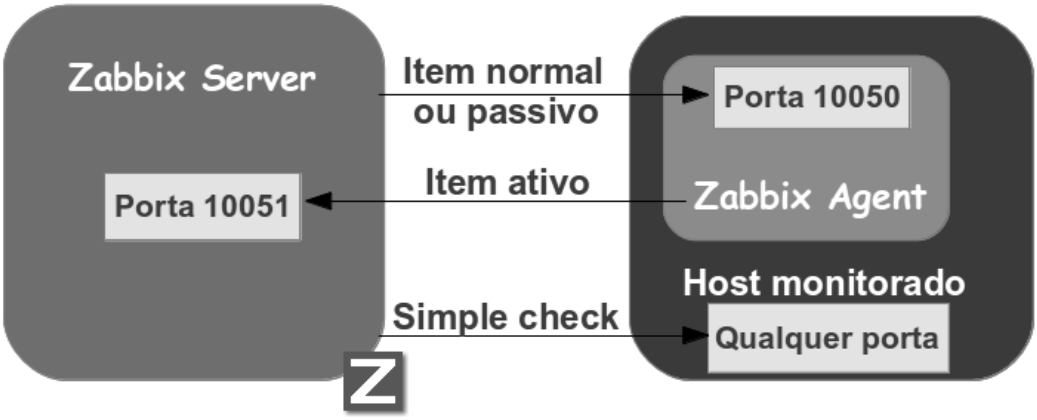


Figura 2 - Comunicação entre o agente e o server Zabbix

Fonte: Adaptado de Alexei Vladishev (http://www.slideshare.net/andredeo/alexei-vladishev-opensourcemonitoringwithzabbix)

A comunicação entre os componentes server e agente pode ser de forma ativa ou passiva. Na forma passiva, o agente envia os dados ao server sobre demanda, ou seja, só envia os dados de acordo com a intensidade e o tipo de requisição feita pelo server. Na forma ativa,

o agente envia ao server uma lista de itens suportados e o server devolve uma resposta contendo os nomes dos itens dos quais o server deseja ser notificado. A partir daí o agente envia ao server os dados referentes as itens escolhidos.

3 Instalando e configurando os softwares e serviços adjacentes ao Zabbix

Os símbolos "**$**", "**#**","**postgres=#**" e "**psql>**", que precederão os comandos abaixo representam, respectivamente, o prompt de comando do usuário comum, do root e do sistema de banco de dados PostgreSQL.

Para instalar os softwares e serviços adjacentes ao Zabbix no Ubuntu 12.04, execute os comandos abaixo.

```
$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo add-apt-repository ppa:ondrej/php5
$ sudo add-apt-repository ppa:pitti/postgresql
$ sudo apt-get update
$ sudo apt-get install -y make flex gcc gpp apache2 apache2-utils ssl-cert php5 php5-pgsql
postgresql-9.2 libapache2-mod-php5 php5-gd php-net-socket postgresql-client libpq5 libpq-
dev  snmp libiksemel-dev libcurl4-gnutls-dev vim libssh2-1-dev libssh2-1 libopenipmi-dev
libsnmp-dev oracle-java7-installer curl fping
```

Ao adicionar os repositórios "ppa:flexiondotorg/java", "ppa:ondrej/php5" e "ppa:pitti/postgresql" pode aparecer um aviso que você está adicionando um repositório do tipo PPA (Personal Package Archives) e a tecla ENTER deve ser pressionada para continuar.

3.1 Configurando o banco de dados no PostgreSQL

Edite o arquivo **/etc/postgresql/9.2/main/pg_hba.conf** e configure o arquivo como mostrado abaixo.

Antes:

```
local   all         postgres                peer
local   all         all                       peer
host    all         127.0.0.1/32            md5
```

Depois:

```
local   all         postgres                trust
local   all         all                       trust
host    all         127.0.0.1/32            trust
```

Salve as alterações e edite outro arquivo **/etc/postgresql/9.2/main/postgresql.conf** conforme mostrado abaixo.

Antes:

```
#listen_addresses = 'localhost'
```

Depois:

```
listen_addresses = 'localhost'
```

Salve as alterações e reinicie o PostgreSQL usando o comando abaixo.

```
$ sudo /etc/init.d/postgresql restart
```

Crie o banco de dados zabbix, usando os comandos abaixo.

```
$ psql -U postgres
postgres=# create database zabbix;
postgres=# \q;
```

Crie no sistema operacional, o usuário a ser usado pelo Zabbix.

```
$ sudo adduser zabbix --shell /bin/false
```

Crie o usuário zabbix no PostgreSQL para gerenciar o banco de dados criado acima.

```
$ sudo -u postgres createuser -a -d -E -P zabbix
```

As senhas do usuário zabbix criado no sistema operacional e no PostgreSQL podem ser diferentes.

3.2 Configurando o PHP

Edite o arquivo **/etc/php5/apache2/php.ini.** Remova o símbolo "**;**" que porventura estiver no início da linha de cada parâmetro abaixo e atribua os seguintes valores em negrito.

date.timezone = **"America/Brasília"**
max_execution_time = **300**
max_input_time = **300**
post_max_size = **16M**

;As opções abaixo com o valor off desabilitam a exibição dos erros para evitar ataque de path disclosered
display_errors = **Off**
display_startup_errors = **Off**

;Desabilita funções que não são usadas pelo Zabbix
disable_functions = **pcntl_alarm, pcntl_fork, pcntl_waitpid, pcntl_wait, pcntl_wifexited, pcntl_wifstopped, pcntl_wifsignaled, pcntl_wexitstatus, pcntl_wtermsig, pcntl_wstopsig, pcntl_signal, pcntl_signal_dispatch, pcntl_get_last_error, pcntl_strerror, pcntl_sigprocmask, pcntl_sigwaitinfo, pcntl_sigtimedwait, pcntl_exec, pcntl_getpriority, pcntl_setpriority, proc_open, popen, disk_free_space, leak, tempfile, exec, system, shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source, apache_get_modules,apache_get_version,apache_getenv,apache_note, apache_setenv, disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore, openlog, proc_nice, symlink, phpinfo**

;Desabilita a exibição da versão do PHP
expose_php = **Off**

;A opção allow_url_fopen permite abrir ou processar uma página ou arquivo externo dentro do script
;php. Embora ela seja uma função útil, usada por scripts que geram uma lista de links a partir de um
;feed, por exemplo, ela pode ser usada para diversos tipos de abusos.

allow_url_fopen = **Off**
allow_url_include = **Off**

;Essa opção foi comentada para não exibir os erros e avisos de algumas funções do PHP
;error_reporting = **E_ALL & ~E_NOTICE**

; Limite de memória a ser usado por um script. O tamanho sugerido abaixo não é o ideal e deve ser ajustado de acordo com o uso e os recursos de memória que tem disponível na máquina em que está sendo instalado o Zabbix.
memory_limit = **1024M**

; Tamanho máximo de um arquivo a ser enviado. O tamanho sugerido abaixo não é o ideal e deve ser ajustado de acordo com o uso e os recursos de armazenamento que tem disponível na máquina em que está sendo instalado o Zabbix.
upload_max_filesize = **1024M**

Salve as alterações e reinicie o Apache usando o comando abaixo.

$ sudo /etc/init.d/apache2 restart

3.3 Configurando o Apache

Editado o arquivo **/etc/apache2/conf.d/security** e altere os parâmetros abaixo informando os valores em negrito.

Nao deixa o Apache exibir informações sobre a versão e módulos instalados em cabeçalhos HTTP.
ServerTokens **Prod**
Nao deixa o Apache exibir informações sobre a versão e módulos instalados em páginas de erro.
ServerSignature **Off**
HTTP TRACE é usado para ecoar todas as informações recebidas. Desabilitando a opção abaixo, o Apache não exibirá cookies HTTP que podem ser usados para roubar sessões HTTP. O HTTP Trace também pode ser vulnerável a ataque do tipo Cross Site Script.
TraceEnable **Off**

Salve as alterações e edite o arquivo **/etc/apache2/apache2.conf** adicionando a linha abaixo ao final do arquivo.

Include /etc/apache2/conf.d/security

A indexação de diretório é uma característica encontrada no Apache por padrão. Quando a indexação de diretório está ativada, é exibida uma lista de arquivos encontrados em cada diretório publica em que não existe um arquivo de indexação (por exemplo index.php, index.html, etc). Isto permite que o usuário visualize arquivos de configuração e backup que possivelmente estejam no diretório e que não deveriam ser visualizados diretamente pelo usuário. Para desabilitar a indexação automática, use os comandos abaixo.

$ sudo a2dismod autoindex
$ sudo /etc/init.d/apache2 restart

Desabilite também o módulo cgi para evitar que scripts cgi possam ser executados no servidor Apache. Não há necessidade de ter o módulo cgi habilitado para o funcionamento do Zabbix. Para desabilitar o módulo cgi, use os comandos abaixo.

```
$ sudo a2dismod cgid
$ sudo /etc/init.d/apache2 restart
```

3.3.1 Ativando o SSL no Apache

Habilite o modo Secure Sockets Layer (SSL) no Apache com os comandos abaixo.

```
$ sudo a2enmod ssl
$ sudo /etc/init.d/apache2 restart
```

Crie o diretório que armazenará o certificado do Apache.

```
$ sudo mkdir /etc/apache2/ssl
```

Crie o certificado SSL para o Apache com validade de 1 ano usando os comandos abaixo.

```
$ cd /etc/apache2/ssl
$ sudo openssl req -newkey rsa:1024 -x509 -nodes -out server.pem -keyout server.pem -days 365
```

Informade os dados do certificado a ser gerado conforme o exemplo abaixo.

Country Name (2 letter code) [AU]:**BR**
State or Province Name (full name) [Some-State]:**Paraiba**
Locality Name (eg, city) []:**Joao Pessoa**
Organization Name (eg, company) [Internet Widgits Pty Ltd]: **Minha Empresa**
Organizational Unit Name (eg, section) []:**STI**
Common Name (eg, YOUR name) []:**nome_servidor.empresa.com.br**
Email Address []:**<enter>**

Com isso foi gerado o arquivo **/etc/apache2/ssl/server.pem** contendo o certificado a ser usado pelo Apache.

Habilite o redirecionamento de URL no Apache, usando os comandos abaixo.

```
$ sudo sudo a2enmod rewrite
$ sudo /etc/init.d/apache2 restart
```

3.4 Instalando o Zabbix

Hoje (20/06/2013) a versão mais recente do Zabbix é a **2.0.6**. Para instalá-la é preciso baixar e compilar o código fonte. Use os comandos abaixo para obter e descompactar o pacote que contém o código fonte e outros arquivos de instalação.

```
$ wget http://downloads.sourceforge.net/project/zabbix/ZABBIX%20Latest%20Stable/2.0.6/zabbix-2.0.6.tar.gz
$ tar xzvf zabbix-2.0.6.tar.gz
```

O pacote de instalação do Zabbix será armazenado no diretório atual (veja em qual diretório está, usando o comando **pwd**). Após a descompactação do pacote será criado o diretório **zabbix-2.0.6** contendo os arquivos de instalação.

Execute os comandos abaixo para popular o banco de dados.

```
$ cat zabbix-2.0.6/database/postgresql/schema.sql | psql -U zabbix zabbix
$ cat zabbix-2.0.6/database/postgresql/images.sql | psql -U zabbix zabbix
$ cat zabbix-2.0.6/database/postgresql/data.sql | psql -U zabbix zabbix
```

Compile e instale o Zabbix executando os comandos abaixo.

```
$ cd zabbix-2.0.6
$ sudo ./configure --enable-server --enable-agent --enable-java --with-postgresql --with-net-snmp  --with-jabber  --with-libcurl=/usr/bin/curl-config --with-ssh2 --with-openipmi
$ sudo make install
$ cd -
```

O significado de cada parâmetro de compilação é mostrado abaixo.

- **--enable-server**: habilita o Zabbix server;
- **--enable-agent**: habilita o agente Zabbix;
- **--enable-java**: habilita o Java gateway, um novo componente do Zabbix 2.0 usado para monitorar aplicações Java;
- **--with-postgresql**: informa que será usado o banco de dados PostgreSQL;
- **--with-net-snmp**: habilita o monitoramento SNMP;
- **--with-jabber**: habilita o envio de alertar via Jabber;
- **--with-libcurl=/usr/bin/curl-config**: habilita o uso da biblioteca curl, usada no monitoramento de aplicações Web. Opcionalmente você pode informar a localização do comando curl-config (use o comando "**whereis curl-config"** para descobrir o caminho);
- **--with-ssh2**: habilita o monitoramento via SSH;
- **--with-openipmi**: habilita o monitoramento de equipamentos via IPMI (http://goo.gl/OX4ui).

Para conhecer mais opções de compilação, execute o comando "**./configure --help**" dentro do diretório de instalação do Zabbix.

3.4.1 Configurando o Zabbix

Os arquivos de configuração do Zabbix 2.0 ficam em **/usr/local/etc**.

Edite o arquivo **/usr/local/etc/zabbix_agentd.conf** e altere os parâmetros abaixo informando os valores que estão em negrito.

PidFile=**/tmp/zabbix_agentd.pid**
LogFile=**/tmp/zabbix_agentd.log**
LogFileSize=**2**
DebugLevel=**3**
Server=**127.0.0.1**
ListenPort=**10050**
Hostname=**informe o nome exato do host, do jeito que aparece no prompt de comandos antes dos símbolos "$" ou "#"**

O parâmetro **LogFileSize** significa o tamanho máximo que o arquivo de log pode ter em mega byte (MB).

Edite o arquivo **/usr/local/etc/zabbix_server.conf** e altere os parâmetros abaixo informando os valores que estão em negrito.

ListenPort=**10051**
LogFile=**/tmp/zabbix_server.log**
LogFileSize=**2**
PidFile=**/tmp/zabbix_server.pid**
DBHost=**localhost**
DBName=**zabbix**
DBUser=**zabbix**
DBPassword=**senha do zabbix para acessar o banco de dados**
StartIPMIPollers=**1**
StartDiscoverers=**5**
Timeout=**3**
FpingLocation=**/usr/bin/fping**

O parâmetro **StartIPMIPollers** só precisa ser configurado se o Zabbix for compilado com a opção **--with-openipmi.**

3.4.2 Criando os scripts de inicialização do Zabbix

Crie os scripts abaixo para que o Zabbix inicie automaticamente junto com a inicialização (*boot*) do sistema operacional.

Crie arquivo **/etc/init.d/zabbix-server** e adicione o conteúdo abaixo.

```
#!/bin/sh
#
# Zabbix daemon start/stop script.
#
# Written by Alexei Vladishev <alexei.vladishev@zabbix.com>.

NAME=zabbix_server
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/home/zabbix/bin
DAEMON=/usr/local/sbin/${NAME}
DESC="Zabbix server daemon"
PID=/tmp/$NAME.pid

test -f $DAEMON || exit 0
set -e
case "$1" in
  start)
        echo "Starting $DESC: $NAME"
        start-stop-daemon --oknodo --start --pidfile $PID \
                --exec $DAEMON
        ;;
  stop)
        echo "Stopping $DESC: $NAME"
        start-stop-daemon --oknodo --stop --pidfile $PID \
                --exec $DAEMON
        ;;
```

```
  restart|force-reload)
        $0 stop
        $0 start
        ;;
  *)
        N=/etc/init.d/$NAME
        echo "Usage: $N {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac
exit 0
```

Crie o arquivo **/etc/init.d/zabbix-agentd** e adicione o conteúdo abaixo.

```
#!/bin/sh
#
# Zabbix agent start/stop script.
#
# Written by Alexei Vladishev <alexei.vladishev@zabbix.com>.

NAME=zabbix_agentd
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/home/zabbix/bin
DAEMON=/usr/local/sbin/${NAME}
DESC="Zabbix agent daemon"
PID=/tmp/$NAME.pid

test -f $DAEMON || exit 0
set -e
case "$1" in
  start)
        echo "Starting $DESC: $NAME"
        start-stop-daemon --oknodo --start --pidfile $PID \
                --exec $DAEMON
        ;;
  stop)
        echo "Stopping $DESC: $NAME"
        start-stop-daemon --oknodo --stop  --pidfile $PID \
                --exec $DAEMON
        ;;
  restart|force-reload)
        $0 stop
        $0 start
        ;;
  *)
        N=/etc/init.d/$NAME
        echo "Usage: $N {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac
exit 0
```

Dê permissão de execução a estes arquivos com o comando abaixo.

```
$ sudo chmod +x /etc/init.d/zabbix-server /etc/init.d/zabbix-agentd
```

Execute os scripts para iniciar os componentes server e agent do Zabbix.

```
$ sudo /etc/init.d/zabbix-server start
$ sudo /etc/init.d/zabbix-agentd start
```

Habilite os scripts para serem executados quando o computador for ligado.

```
$ sudo update-rc.d -f zabbix-server defaults
$ sudo update-rc.d -f zabbix-agentd defaults
```

3.4.3 Habilitando a interface web do Zabbix no Apache usando SSL

Copie os arquivos de frontend do Zabbix para o diretório **/var/www/zabbix**, executando os comandos abaixo.

```
$ sudo  mkdir /var/www/zabbix
$ sudo cp -R zabbix-2.0.6/frontends/php/*  /var/www/zabbix/
$ sudo chown -R www-data:www-data /var/www/zabbix/
```

Para habilitar o acesso a interface web do Zabbix com SSL, localizada em **/var/www/zabbix,** remova os arquivos **/etc/apache2/sites-available/default e /etc/apache2/sites-available/default-ssl.** Crie novamente o arquivo **/etc/apache2/sites-available/default** e adicione as linhas abaixo. No exemplo abaixo, os IPs 127.0.0.1, 192.168.1.1/24 e 192.168.2.0/24 correspondem, respectivamente, ao endereço localhost do servidor Zabbix, um IP específico da rede interna e a uma faixa de rede que serão os únicos a poderem acessar o Zabbix. Este IPs podem ser substituídos por outros de forma que o acesso ao Zabbix seja restringido.

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www
        <Directory />
                Deny from all
        </Directory>
        <Directory /var/www/>
                Options +FollowSymLinks
                AllowOverride None
                Deny from all
                Allow from 127.0.0.1 192.168.1.1/24 192.168.2.0/24
       </Directory>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/
```

```
    <Directory />
        Deny from all
    </Directory>

    #********** Zabbix **********#
    Alias /zabbix /var/www/zabbix
    <Directory /var/www/zabbix>
        Options +FollowSymLinks
        AllowOverride None
        Deny from all
        Allow from 127.0.0.1 192.168.1.1/24 192.168.2.0/24
    </Directory>
    #************************************#

    ErrorLog /var/log/apache2/error.log
    LogLevel warn
    CustomLog /var/log/apache2/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/server.pem
</VirtualHost>
```

No final deste mesmo arquivo, após a seção referente a porta 443, adicione as linhas abaixo para criar o redirecionamento de páginas. Quando o usuário tentar acessar o Zabbix via HTTP, o Zabbix será exibido usando HTTPS. Nas linhas abaixo ip-servidor deve ser substituído pelo endereço IP do computador em que o Zabbix está sendo instalado.

```
<IfModule mod_rewrite.c>
    <IfModule mod_ssl.c>
        <LocationMatch /zabbix>
            RewriteEngine on
            RewriteCond %{HTTPS} !^on$ [NC]
            RewriteRule . https://ip-servidor/%{REQUEST_URI} [L]
        </LocationMatch>
    </IfModule>
</IfModule>
```

Salve as alterações e reinicie o Apache usando o comando abaixo.

```
$ sudo /etc/init.d/apache2 restart
```

3.4.4 Acessando a interface web do Zabbix

Usando um navegador acesse o Zabbix no endereço https://ip-do-servidor/zabbix e siga as próximas recomendações.

Na **Figura 3** clique no botão **Next.**

Figura 3 - Tela inicial de configuração do Zabbix

Na Figura 4 cheque as dependências do Zabbix. Se estiver tudo OK, clique em **Next**.

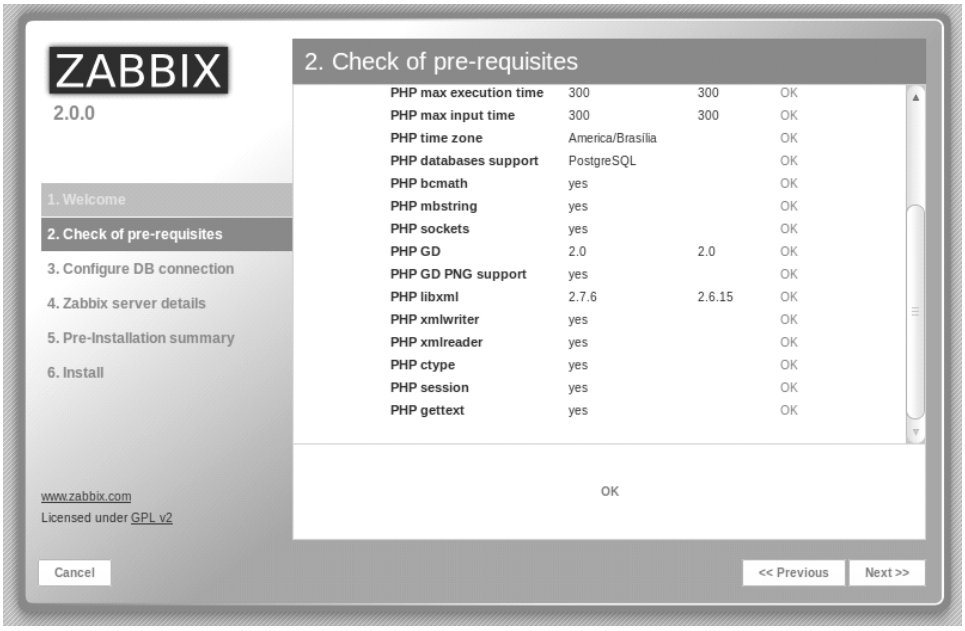


Figura 4 - Checando as dependências do Zabbix

Caso contrário, reveja os passos executados ao longo deste artigo para encontrar o problema ou peça ajuda na lista de usuários brasileiros do Zabbix em http://br.groups.yahoo.com/group/zabbix-brasil.

Na Figura 5 Informe o tipo da base de dados, o usuário e a senha. Em seguida, clique no botão **Test Connection**. Se estiver OK, clique em Next.

Figura 5 - Configurando a conexão com o banco de dados

Na Figura 6 informe o IP do servidor Zabbix e a porta em que ele será executado (a padrão é 10051). No campo **Name** informe o nome do computador em que o Zabbix está instalado (isto é útil quando se administra vários servidores Zabbix). Depois clique em **Next**.

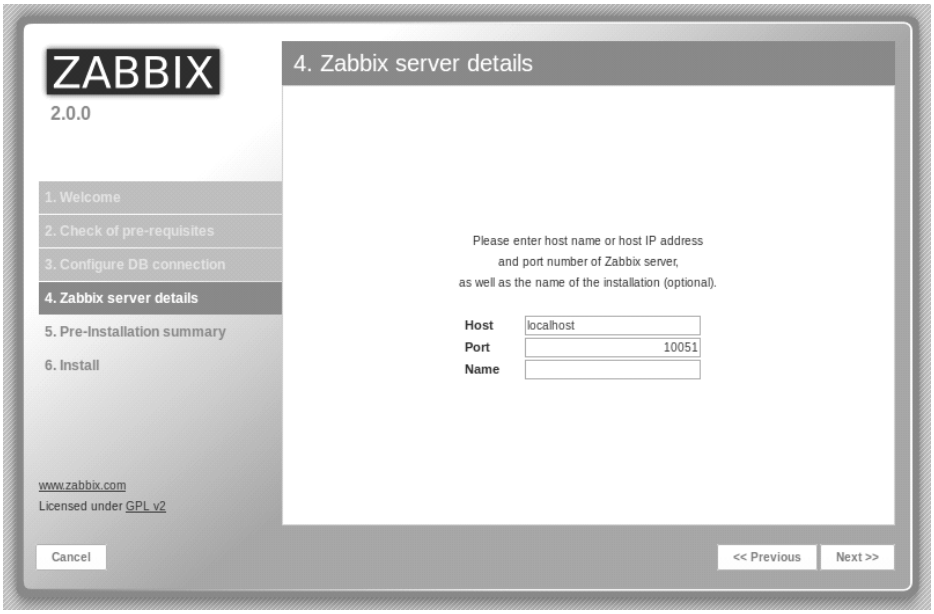


Figura 6 - Informando detalhes do servidor Zabbix

Na Figura 7 revise as configurações e se estiver ok, clique em **Next**.

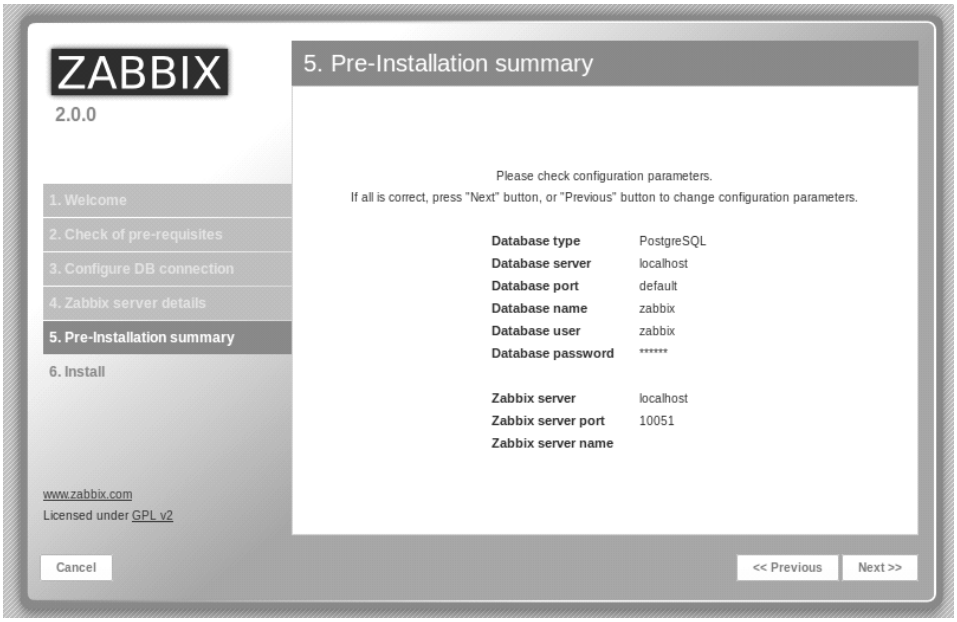


Figura 7 - Resumo da instalação

Na Figura 8 clique em **Finish**. Se nesta tela for exibido um erro de permissão durante a atualização do arquivo de configuração, cheque a permissão do diretório **/var/www/zabbix** e configure da forma mostrada na seção 3.4.3 deste artigo.

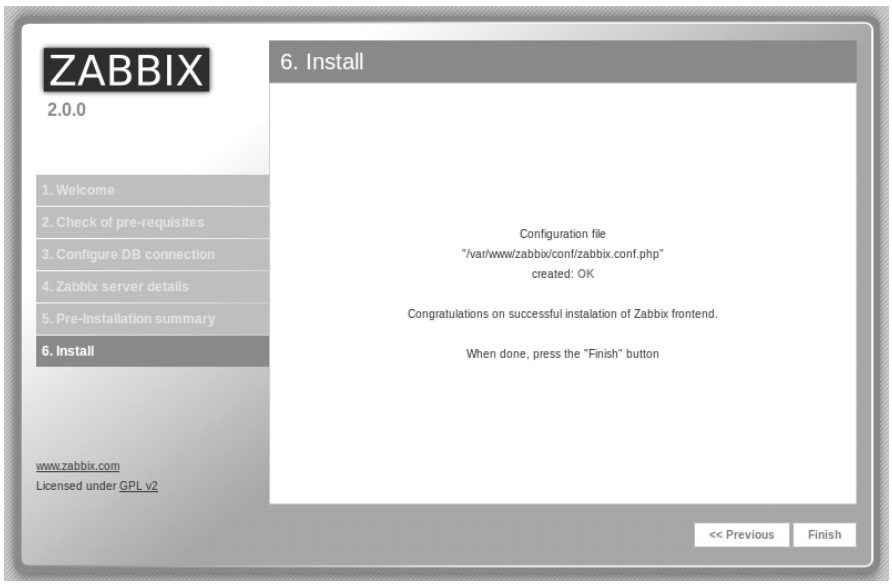


Figura 8 - Finalizando a instalação

Pronto! O Zabbix está instalado. Faça login no Zabbix com o usuário **Admin** e senha **zabbix**.

Acesse o menu **Configuration > Hosts.** Como mostra a Figura 9, o status do host Zabbix server é **Not monitored** (link na cor vermelha)**.**

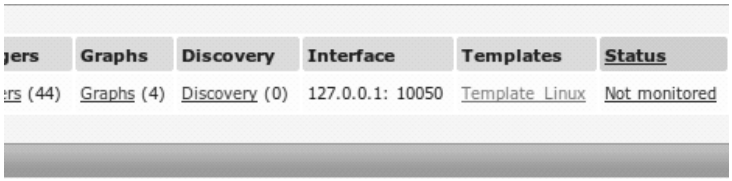


Figura 9 - Status do servidor Zabbix

Clique sobre o link **Not monitored** para habilitar o monitoramento. Será exibido a caixa mostrada na Figura 10.

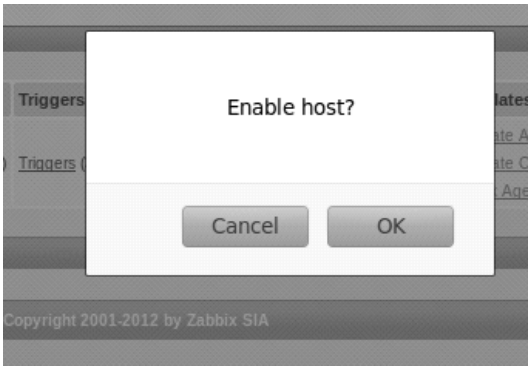


Figura 10 - Habilitando o monitoramento do servidor Zabbix

Clique em OK para habilitar o monitoramento. Perceba que o status será alterado para **Monitored** (link na cor verde).

Depois acesse o menu **Monitoring > Dashboard** e veja que na linha **Zabbix server is running,** o valor é **Yes,** o que significa que o componente Zabbix-server está sendo executado. Veja a Figura 11.

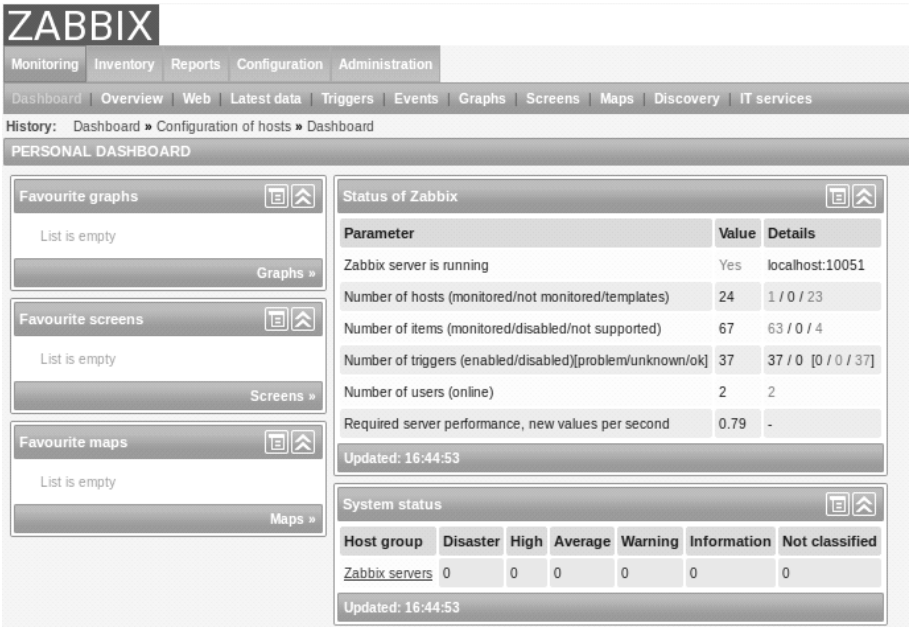
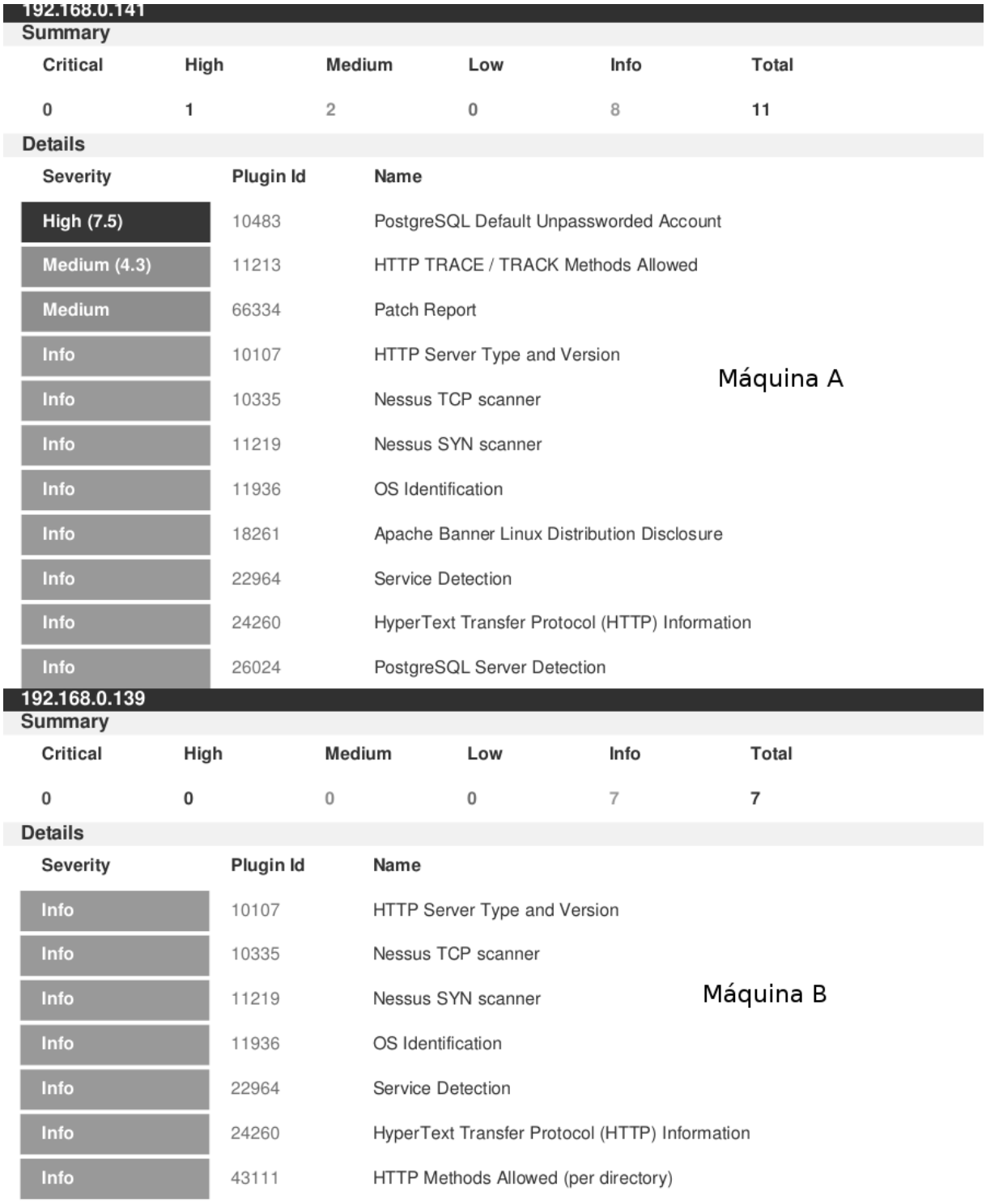


Figura 11 - O componente server do Zabbix está sendo executado

4. Analisando as vulnerabilidade do ambiente

Para efeitos de comparação de vulnerabilidades existentes no ambiente, foi instalado o Zabbix e os serviços adjacentes em uma máquina virtual, chamada A e com o IP 192.168.0.141, com nenhuma preocupação com a segurança. Em outra máquina virtual, chamada B e com o IP 192.168.0.139, foi instalado o Zabbix e os serviços adjacentes seguindo as intruções deste artigo. Para procurar por vulnerabilidades nos ambientes foi utilizado o *software* Nessus 5.2.1 (http://www.tenable.com/products/nessus) instalado em uma terceira máquina com o IP 192.168.0.81. As duas máquinas virtuais que possui o Zabbix instalado possuem as mesmas configurações de *hardware* mostradas no início do artigo.

As vulnerabilidades encontradas pelo Nessus em cada máquina virtual estão listadas na Figura 12.



192.168.0.141

Summary

Critical	High	Medium	Low	Info	Total
0	1	2	0	8	11

Details

Severity	Plugin Id	Name
High (7.5)	10483	PostgreSQL Default Unpassworded Account
Medium (4.3)	11213	HTTP TRACE / TRACK Methods Allowed
Medium	66334	Patch Report
Info	10107	HTTP Server Type and Version
Info	10335	Nessus TCP scanner
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	18261	Apache Banner Linux Distribution Disclosure
Info	22964	Service Detection
Info	24260	HyperText Transfer Protocol (HTTP) Information
Info	26024	PostgreSQL Server Detection

Máquina A

192.168.0.139

Summary

Critical	High	Medium	Low	Info	Total
0	0	0	0	7	7

Details

Severity	Plugin Id	Name
Info	10107	HTTP Server Type and Version
Info	10335	Nessus TCP scanner
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	22964	Service Detection
Info	24260	HyperText Transfer Protocol (HTTP) Information
Info	43111	HTTP Methods Allowed (per directory)

Máquina B

Figura 12 - Vulnerabilidades encontradas nas máquinas virtuais.

Como pode ser visto na Figura 12, o Nessus conseguiu encontrar menos vulnerabilidades na máquina B.

As vulnerabilidades *PostgreSQL Default Unapassworded Account* e *Patch Report* se referem ao PostgreSQL que estava configurado na máquina A para receber conexões de qualquer máquina da rede 192.168.0.0/24. Na máquina B, o PostgreSQL foi configurado para receber apenas conexões locais, ou seja, do IP 127.0.0.1. Isso reforça a importância de limitar

o acesso remoto ao banco de dados conforme mostrado na configuração dos arquivos pg_hba.conf e postgresql.conf mostrado na seção 3.1

A vulnerabilidade *HTTP TRACE/TRACK Methods Allowed* encontrada pelo Nessus na máquina A, foi devido a configuração do parâmetro **TraceEnable** do Apache que, por padrão, estava com o valor **On.** Como na máquina B, este parâmetro foi ajustado para o valor **Off,** o Nessus não conseguiu encontrar essa vulnerabilidade.

O Nessus também conseguiu obter menos informações do servidor Web na máquina em que foram aplicadas as configurações de segurança. Um exemplo pode ser visto na Figura 13.

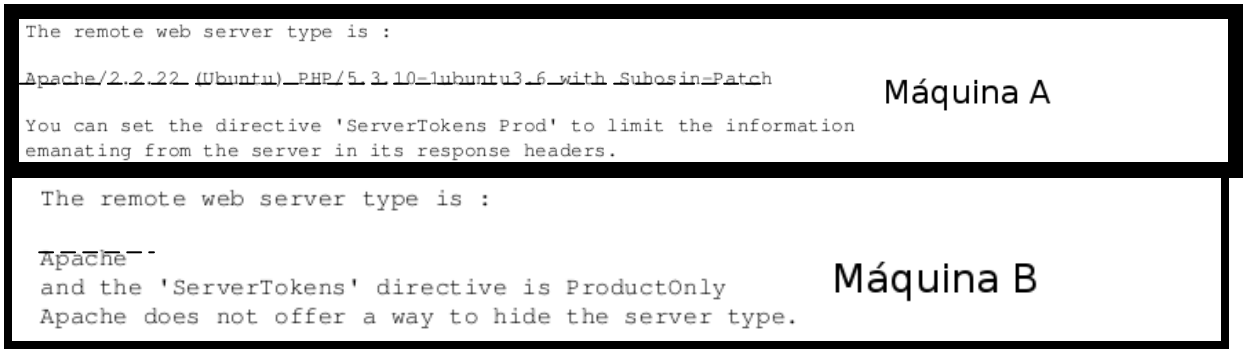


Figura 13 - Obtendo informações do servidor Web.

A Figura 13 mostra que ao analisar o item *HTTP Server Type and Version* em cada máquina virtual, o Nessus conseguiu obter menos informações acerca do servidor Web. Na máquina A, o parâmetro **ServerTokens** do Apache estava com o valor **Full** e na máquina B estava com o valor **Prod**.

Considerações Finais

Ao longo do artigo foi mostrada a instalação do Zabbix aplicando configurações de segurança aos serviços adjascentes. As configurações de segurança mostradas aqui são básicas e bastante conhecidas, mas não são aplicadas em todos os ambiente de produção.

Além dessas configurações também é fortemente recomendado aplicar configurações de segurança ao sistema operacional e remover pacotes e serviços que não são usados, mas que são instalados por padrão. Também é recomendado usar o módulo modsecurity do Apache que funciona como um firewall de aplicações Web ajudando a melhorar o nível de segurança do Zabbix.

Outra necessidade importante é analisar minunciosamente o código fonte do Zabbix a fim de indentificar pontos vulneráveis para reportar a equipe de desenvolvimento para que assim medidas de prevenção sejam aplicadas no desenvolvimento do Zabbix.

Outra recomendação importante é ficar atento as atualizações dos softwares instalados no servidor. Na maioria das vezes essas atualizações trazem melhorias de segurança.

A comunicação entre os componentes agente e server e entre proxy e server não é criptografada. Todas as informações são trafegadas na rede em texto plano e estão sugeitas a interceptação. Para melhorar a segurança desta comunicação fica o desafio de usar criptografia sem impactar muito na comunicação entre os componentes afetando negativamente na coleta dos dados e no monitoramento dos equipamentos, principalmente os que não se encontram na rede interna e que são monitorados usando um link de baixa taxa de transmissão de dados.

Outro projeto futuro é testar o uso do Zabbix com um firewall a nível de banco de dados, como por exemplo o GreenSQL (www.greensql.com), que ajuda a minimizar os ataques de SQL Injecton.

Referências

FOROUZAN, Behrouz A. Comunicação de Dados e Redes de Computadores. 3. ed. Porto Alegre. Bookman, 2006.

OLUPS, Rihard. Zabbix 1.8 Networking Monitoring. Birmingham. Packt Publishing, 2010.

RIGGS, Simon. PostgreSQL 9 Administration Cookbook . Birmingham. Packt Publishing, 2010.

VALADE, Janete. PHP5 for Dummies. Indianapolis. Wiley Publishing, Inc, 2004.

WIKI APACHE. What is Apache? Disponível em: <http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F> Acesso em: 26 jun. de 2013.

Bibliografia Consultada

10 Apache Security and Hardening Tips. Disponível em: <http://www.kyplex.com/docs/apache-security.html>. Acesso em: 26 jun. de 2013.

Apache 2.0 Hardening Guide. Disponível em: <http://xianshield.org/guides/apache2.0guide.html>. Acesso em: 26 jun. de 2013.

Apache Tips: Disable the HTTP TRACE Method Disponível em: <http://www.ducea.com/2007/10/22/apache-tips-disable-the-http-trace-method>. Acesso em: 26 jun. de 2013.

DEO, André e PIRES, Aécio. Gerência de Redes com Zabbix. Revista Espírito Livre Ed. 18 págs. 69 a 73. Disponível em: <http://www.revista.espiritolivre.org/?p=693> Acesso em: 26 jun. de 2013.

Hide Apache ServerSignature/ServerTokens/PHP X-Powered-By. Disponível em: <http://www.if-not-true-then-false.com/2009/howto-hide-and-modify-apache-server-information-serversignature-and-servertokens-and-hide-php-version-x-powered-by/> . Acesso em: 26 jun. de 2013.

KLEIN, Amit. Cross site scripting explained. Disponível em: <http://crypto.stanford.edu/cs155/papers/CSS.pdf>. Acesso em: 26 jun. de 2013.

LAVLU, Ibrahim. How to enable mod_rewrite in apache2.2 (debian/ubuntu). Disponível em: <http://www.lavluda.com/2007/07/15/how-to-enable-mod_rewrite-in-apache22-debian/ >. Acesso em: 26 jun. de 2013.

MORIMOTO, Carlos E. Dicas de segurança para o PHP em um servidor LAMP Disponível em: <http://www.guiadohardware.net/dicas/seguranca-php-lamp.html>. Acesso em: 26 jun. de 2013.

Manual do Zabbix 2.0. Disponível em: <http://www.zabbix.com/documentation/2.0/manual/introduction>. Acesso em: 26 jun. de 2013.

PIRES, Diego. WEB HACKING: Ataques e Vulnerabilidades em Aplicações Web. Disponível em: <http://blog.corujadeti.com.br/web-hacking-ataques-e-vulnerabilidades-em-aplicacoes-web/>. Acesso em: 26 jun. de 2013.

STUTTARD, Dafydd. PINTO, Marcus. The Web Application Hacker's Handbook : Discovering and Exploiting Security Flaws . Indianapolis. Wiley Publishing, Inc, 2008.