Bacula Enterprise Security and Threat Analysis

Bacula Systems Documentation

Contents

1 Protection **2**
- 1.1 General Considerations . . . 2
- 1.2 Running as Non-Root . . . 3
- 1.3 TLS - Communication Encryption . . . 4
- 1.4 Data Encryption . . . 4
- 1.5 Migration and Copy Jobs . . . 4

2 Identification **4**
- 2.1 OSSEC - Wazuh - SIEM Integration . . . 4

3 Detection **5**
- 3.1 Malware Detection . . . 5

4 Recovery **7**

Contents

The following article aims at presenting information on security and threat analysis.

1 Protection

1.1 General Considerations

Important: Security means being able to restore your files, so read the Critical Items Chapter chapter.

- The Clients (**bacula-fd**) must run as root to be able to access all the system files.
- It is not necessary to run the Director as `root`.
- It is not necessary to run the Storage daemon as `root`, but you must ensure that it can open the tape drives, which are often restricted to `root` access by default. In addition, if you do not run the Storage daemon as `root`, it will not be able to automatically set your tape drive parameters on most `OS`es since these functions, unfortunately require root access.
- You should restrict access to the Bacula configuration files, so that the passwords are not world-readable. The Bacula daemons are password protected using CRAM-MD5 (i.e. the password is not sent across the network). This will ensure that not everyone can access the daemons. It is a reasonably good protection, but can be cracked by experts.
- If you are using the recommended ports 9101, 9102, and 9103, you will probably want to protect these ports from external access using a firewall and/or using tcp wrappers (**etc/hosts.allow**).
- By default, all data that is sent across the network is encrypted. Read the TLS (SSL) Communications Encryption section.
- You should ensure that the Bacula working directories are readable and writable only by the Bacula daemons.
- If you are using **MySQL** it is not necessary for it to run with **root** permission.

- The default Bacula **grant-mysql-permissions** script grants all permissions to use the database without a password. It is recommended to change that.
- Mind that Bacula is a network program, so anyone anywhere on the network with the console program and the Director's password can access Bacula and the backed up data.
- You can restrict what IP addresses Bacula will bind to by using the appropriate **DirAddress**, **FDAddress**, or **SDAddress** records in the respective daemon configuration files.
- Be aware that if you are backing up your database using the default script, if you have a password on your database, it will be passed as a command line option to that script, and any user will be able to see this information. If you want it to be secure, you will need to pass it by an environment variable or a secure file.

Note: For more details, read Backing Up Your Bacula Database - Security Considerations.

1.2 Running as Non-Root

It is a good idea to run daemons with the lowest possible privileges. In other words, if you can, don't run applications as root which do not have to be root. The Storage Daemon and the Director Daemon do not need to be root. The File Daemon needs to be root in order to access all files on your system. In order to run as non-root, you need to create a user and a group. Choosing `bacula` as both the user name and the group name sounds like a good idea to me.

The FreeBSD port creates this user and group for you. Here is what those entries looked like on my FreeBSD laptop:

```
bacula:*:1002:1002::0:0:Bacula Daemon:/var/db/bacula:/sbin/nologin
```

I used **vipw** to create this entry. I selected a User ID and Group ID of 1002 as they were unused on my system.

I also created a group in **/etc/group:**:

```
bacula:*:1002:
```

The `bacula` user (as opposed to the Bacula daemon) will have a home directory of **/var/db/bacula** which is the default location for the Bacula database.

Now that you have both a bacula user and a bacula group, you can secure the bacula home directory by issuing this command:

```
chown -R bacula:bacula /var/db/bacula/
```

This ensures that only the bacula user can access this directory. It also means that if we run the Director and the Storage daemon as bacula, those daemons also have restricted access. This would not be the case if they were running as root.

It is important to note that the storage daemon actually needs to be in the operator group for normal access to tape drives etc (at least on a FreeBSD system, that's how things are set up by default) Such devices are normally . It is easier and less error prone to make Bacula a member of that group than it is to play around with system permissions.

Starting the Bacula daemons

To start the Bacula daemons on a FreeBSD system, issue the following command:

```
/usr/local/etc/rc.d/bacula-dir start
/usr/local/etc/rc.d/bacula-sd start
/usr/local/etc/rc.d/bacula-fd start
```

To confirm they are all running:


```
$ ps auwx | grep bacula
root 63418 0.0 0.3 1856 1036 ?? Ss 4:09PM 0:00.00
    /usr/local/sbin/bacula-fd -v -c /usr/local/etc/bacula-fd.conf
bacula 63416 0.0 0.3 2040 1172 ?? Ss 4:09PM 0:00.01
    /usr/local/sbin/bacula-sd -v -c /usr/local/etc/bacula-sd.conf
bacula 63422 0.0 0.4 2360 1440 ?? Ss 4:09PM 0:00.00
    /usr/local/sbin/bacula-dir -v -c /usr/local/etc/bacula-dir.conf
```

1.3 TLS - Communication Encryption

Bacula TLS is built-in network encryption code to provide secure network transport similar to that offered by **stunnel** or **ssh**. The **Bacula** TLS encryption applies only to information transmitted across a network, so the data written to Volumes by the Storage daemon is not encrypted by this code:

- tls

For data encryption, see the PKI options described on the dataencryption page.

1.4 Data Encryption

Bacula permits file data encryption and signing within the File Daemon (or Client) prior to sending data to the Storage Daemon.

Read the article about how to set up the File Daemon to encrypt your data here:

- FileDaemonEncryption

Bacula can also encrypt the data within the Storage Daemon prior to write the volume to disk. The access to the data is then only possible with the appropriate encryption keys. More information can be found here volumeencryption.

1.5 Migration and Copy Jobs

Bacula allows you to copy backup jobs to other locations for a geo-redundant copy or migrate your jobs to a different medium (D2D2T strategy). This will improve the security of your backups in case of a ransomware attack or a disaster scenario.

Refer to Replication: Copy/Migration Jobs for more details.

2 Identification

2.1 OSSEC - Wazuh - SIEM Integration

Security information and event management (SIEM) is a software product that combines services security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by applications and network hardware, they are also used to log security data and generate reports for compliance purposes

The `bacula-enterprise-wazuh-rules` package contains a log decoder and and a set of rules to integrate Bacula into the Wazuh SIEM software. (https://wazuh.com).

Wazuh is based on OSSEC and is used to collect, aggregate, index and analyze security data, helping organizations detect intrusions, threats and behavioral anomalies.

The integration of the Bacula server with the Wazuh console can be done via syslog or via the Wazuh agent.

The syslog configuration can be done via the rsyslog package with a configuration such as:

```
# cat /etc/rsyslog.d/bacula.conf daemon.* @wazuhserver
```

The `bacula-enterprise-wazuh-rules` package that contains the Wazuh rules must be installed on the Wazuh server.

3 Detection

3.1 Malware Detection

Distinction between Antivirus Plugin and Malware

In an Antivirus Check, Bacula will transmit the files to the ClamAV Antivirus Socket, which will perform the scan and report to Bacula if any viruses are discovered. In the instance of Malware, Bacula will retrieve the Malware database signatures from https://abuse.ch/ and then do a file verification with those signatures. If a Backup job finds malware in the backup content, an error message is generated and the Job status is changed.

Bacula allows you to configure your jobs to detect known Malware. The detection can be done at the end of the Backup job and/or with a Verify job.

The Job directive `Check Malware = yes/No` can control the behavior.

```
Job {
  Name = MyBackup
  Check Malware = yes
  Fileset = FullSet
  JobDefs = Default
}

Fileset {
  Name = FullSet
  Include {
     Options {
        Signature = md5
     }
     File = /home
  }
}
```

The Fileset of the Backup Job must use the `Signature = MD5` or `Signature = SHA256` option to use the `Check Malware` directive.

By default, the Malware database is fetched from `abuse.ch`. If needed, it can be adapted with the Director `MalwareDatabaseCommand` directive.

If a Backup job detects a malware in the backup content, an error is reported and the Job status is adapted.

```
 20-Sep 12:26 zog8-dir JobId 9: Start Backup JobId 9, Job=backup.2022-09-20_12.26.30_13
 ...
 20-Sep 12:26 zog8-dir JobId 9: [DI0002] Checking file metadata for Malwares
 20-Sep 12:26 zog8-dir JobId 9: Error: [DE0007] Found Malware(s) on JobIds 9
```

(continues on next page)

(continued from previous page)

```
    Build OS:               x86_64-pc-linux-gnu archlinux
    JobId:                  9
    Job:                    backup.2022-09-20_12.26.30_13
    Backup Level:           Full
    ...
    Last Volume Bytes:      659,912,644 (659.9 MB)
    Non-fatal FD errors:    1
    SD Errors:              0
    FD termination status:  OK
    SD termination status:  OK
    Termination:            Backup OK -- with warnings

The list of the Malware detected in a given Job can be displayed with the ``list files␣
↪type=malware``
command.
```

```
*list files type=malware jobid=1
+-------+------------------------------+---------------+----------+
| jobid | filename                     | description   | source   |
+-------+------------------------------+---------------+----------+
|     1 | /tmp/regress/build/po/fr.po | Malware found | abuse.ch |
+-------+------------------------------+---------------+----------+
```

A Verify Job with the level `VolumeToCatalog` or `Data` can be configured with the `Check Malware=yes` directive to report malware in addition to standard errors detected by the Verify Job feature.

```
*run job=VerifyVolCat jobid=1 yes
Job queued. JobId=7
*wait
You have messages.
*messages
20-Sep 12:26 zog8-dir JobId 7: Verifying against JobId=1 Job=backup.2022-09-20_12.25.48_
↪03
20-Sep 12:26 zog8-dir JobId 7: [DI0002] Checking file metadata for Malwares
20-Sep 12:26 zog8-dir JobId 7: Error: [DE0007] Found Malware(s) on JobIds 1
20-Sep 12:26 zog8-sd JobId 7: Ready to read from volume "TestVolume001" on File device
↪"FileChgr1-Dev1" (/tmp/regress/tmp).
20-Sep 12:26 zog8-sd JobId 7: Forward spacing Volume "TestVolume001" to addr=216
20-Sep 12:26 zog8-sd JobId 7: Elapsed time=00:00:01, Transfer rate=94.08 M Bytes/second
20-Sep 12:26 zog8-dir JobId 7: Bacula zog8-dir 14.1.1 (12Aug22):
  Build OS:               x86_64-pc-linux-gnu archlinux
  JobId:                  7
  Job:                    VerifyVolCat.2022-09-20_12.26.14_09
  Fileset:                Full Set
  Verify Level:           VolumeToCatalog
  Client:                 zog8-fd
  Verify JobId:           1
  Verify Job:
  Start time:             20-Sep-2022 12:26:16
  End time:               20-Sep-2022 12:26:25
  Elapsed time:           9 secs
  Accurate:               yes
```

(continues on next page)

(continued from previous page)

```
 Files Expected:         3,640
 Files Examined:         3,640
 Non-fatal FD errors:    1
 SD Errors:              0
 FD termination status:  OK
 SD termination status:  OK
 Termination:            Verify OK -- with warnings
```

4 Recovery

One of the major goals of Bacula is to ensure that you can restore tapes (the word tape is also used to include disk Volumes) that you wrote years ago. This means that each new version of Bacula should be able to read old format tapes. The first problem you will have is to ensure that the hardware is still working some years down the road, and the second problem will be to ensure that the media will still be good, then your OS must be able to interface to the device, and finally Bacula must be able to recognize old formats. All the problems except the last are ones that we cannot solve, but by careful planning you can.

Since the very beginning of Bacula until today, there have been two major Bacula tape formats.

Though the tape format is fixed, the kinds of data that we can put on the tapes are extensible, and that is how we added new features such as Win32 data, encrypted data, etc. Obviously, an older version of Bacula would not know how to read these newer data streams, but each newer version of Bacula should know how to read all the older streams.

If you want to be 100% sure that you can read old tapes, you should:

- Try reading old tapes from time to time – e.g. at least once a year.
- Keep statically linked copies of every version of Bacula that you use in production then if for some reason, we botch up old tape compatibility, you can always pull out an old copy of Bacula.

