Security Plugin

Bacula Systems Documentation

Contents

Contents

---

1 Overview

1.1 Features Summary

The **Bacula Enterprise Security** plugin provides a framework that can be used to check for vulnerabilities using the Bacula File Daemon on your servers. The security checks are executed once a day during any Backup Job. Information about any vulnerabilities found is printed in the Job report and a potential error message can be logged in the Job log. A *Security Object* will be inserted in the catalog for further analysis.

2 Security Hooks

Security hooks are installed in `/opt/bacula/etc/bcheck_sys.d` and can be executed separately.

2.1 Basic

Linux

```
000-bacula-basic
```

The *basic* check will analyse the Bacula Director configuration to check the password policy. It also controls the different permission checks on various Bacula files under `/opt/bacula`.

Windows

```
001-WindowsUpdate.ps1
```

The *WindowsUpdate* check will analyse the Windows Security updates and report the uninstalled ones with relevant level of importance.

3 Installation

3.1 Packages

Packages of the **Security** plugin are available for supported platforms. Please contact Bacula Systems Support team to get them.

Download the **Security** plugin package to your server where a Bacula File Daemon is installed and then install using the package manager

Debian/Ubuntu

```
dpkg -i bacula-enterprise-security-plugin*.deb
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the **Security** plugin.

Rhat/ Centos

```
rpm -ivh bacula-enterprise-security-plugin*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the **Security** plugin.

Windows

The **Bacula Enterprise Security** plugin is selectable as a component of the **File Daemon** windows installer.

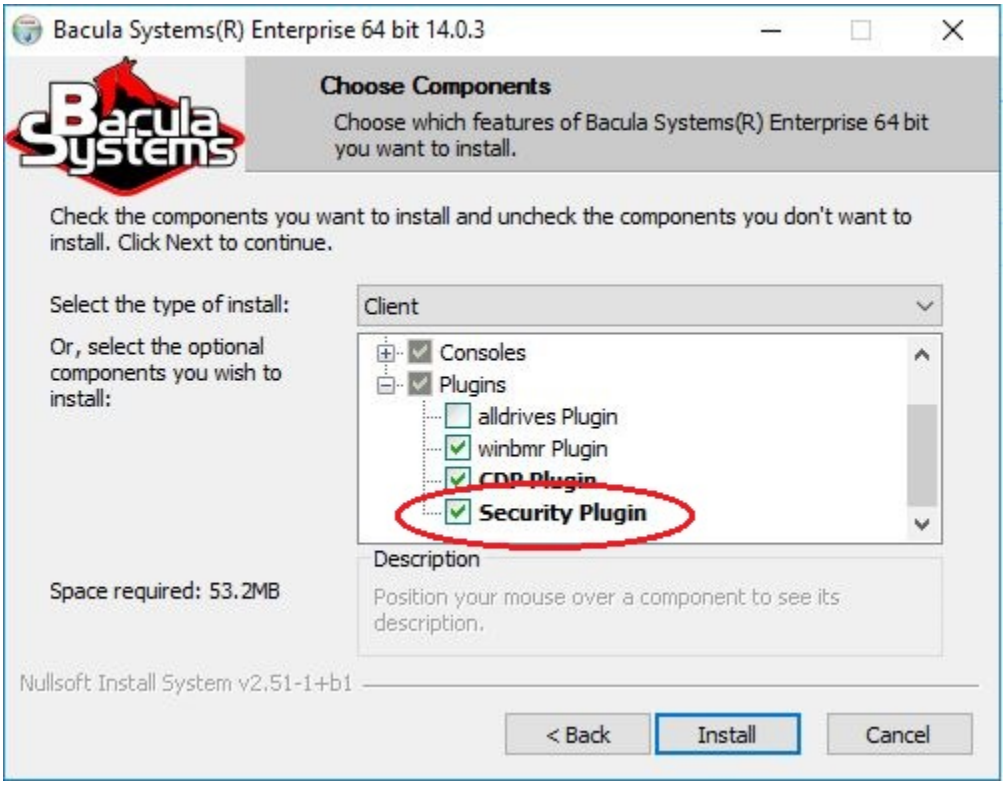

Fig. 1: The Security plugin in the File Daemon windows installer

4 Configuration

4.1 File Daemon Configuration

On the File Daemon host server, the **Plugin Directory** directive of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` has to point to where the `security-fd.so` plugin is installed. The standard directory for Bacula plugins is `/opt/bacula/plugins`

```
FileDaemon {
   Name = bacula-fd
   Plugin Directory = /opt/bacula/plugins
   Plugin Options = "security: interval=2days"
   ...
}
```

The **Plugin Options** directive can be used to configure options of the **Security** plugin.

Table 1: Security plugin parameters

Option	Default	Description
interval	24h	The interval parameter specifies the time between two security checks.

5 Advanced

5.1 Forcing a New Check

It is possible to force a new check by deleting the file `/opt/bacula/working/security.ts`

5.2 Hook Protocol Definition

Security hooks can be written in any language. Some environment variables are passed to all hooks.

Table 2: Environnement variables

Option	Default	Description
BACULA_WORKINGDIR BACULA_SYSCONFDIR BACULA_BINDIR	/opt/bacula/working /opt/bacula/etc /opt/bacula/bin	Bacula Working directory Bacula Configuration directory Bacula Binary directory

The output provided by the hook is a JSON object with the following information:

```
   {
   "source": "chkrootkit",
   "version": "0.52",
   "error": 1,
   "events": [
       {
        "level": 'f',
        "message": "INFECTED: Possible Malicious Linux.Xor.DDoS installed"
       },
       {
        "level": 'f',
        "message": "INFECTED: Possible Malicious Linux.XXX installed"
       }
   ]
},
```

Table 3: JSON fields

Option	Description
source | (String) Name of the hook version | (String) Version of the hook program error | (Int) different from zero to raise an error events | (Array) list of different events	

Each events have the following information

Table 4: JSON Events fields

Option	Description
level message	(char) Status of the test (f: fatal, T: ok, W: warning) (String) Error to be displayed. (contains simple characters)

