

# 发布签名

## 签收方式

LEDE使用GnuPG (<https://www.gnupg.org/>)和*usign*，OpenBSD的衍生物表示 (<https://www.openbsd.org/papers/bsdcan-signify.html>)功能。

**OPKG**软件包管理器 在安装软件包时使用用户 **Ed25519**签名来验证存储库元数据，而发行映像文件通常由具有分离的**GPG**签名的一个或多个开发人员签名，以允许用户验证安装文件的完整性。

我们的签名文件携带扩展 **.sig**，而分离的**GPG**签名结束 **.gpg**。

请注意，并不是每个文件都是单独签名的，而是我们正在 **sha256sums** 为存储库签名 - **Packages** 建立一个信任链的文件：**SHA256**校验和将验证实际文件的完整性，而签名将验证文件的完整性包含校验和。

## 验证下载完整性

为了验证固件下载的完整性，您需要执行以下步骤：

1. 下载 **sha256sum** 和 **sha256sum.gpg** 文件
2. 检查签名 `gpg --with-fingerprint --verify sha256sum.gpg sha256sum`，确保**GnuPG**命令报告良好的签名，并且指纹与我们的指纹页面上列出的指纹匹配。
3. 将固件映像下载到与该 **sha256sums** 文件相同的目录中，并使用以下命令验证其校验和：  
`sha256sum -c --ignore-missing sha256sums`

## 验证脚本

您可以使用我们提供的便捷脚本来自动执行所需的下载和签名验证步骤。

以下是使用该 **download.sh** 脚本的示例脚本。

```
user @ host: ~$ wget -O download.sh https://lede-project.org/_export/code/docs/user-guide/release_signatures?codeblock=1
--2016-12-24 01: 48: 14-- https://lede-project.org/_export/code/docs/user-guide/release_signatures?codeblock=1
解决lede-project.org (lede-project.org) ... 139.59.209.225, 2a03: b0c0: 3: d0 :: 1a
f1: 1
连接到lede-project.org (lede-project.org) | 139.59.209.225 |: 443 ...已连接。
发送HTTP请求, 等待响应... 200 OK
长度: 未指定[text / plain]
保存到: 'download.sh'
```

```
[<=>] 4,091 --.- K / s在0s
```

```
2016-12-24 01:48:14 (722 MB / s) - 'download.sh'saved [4091]
```

```
user @ host: ~$ chmod + x download.sh
```

```
user @ host: ~$ ./download.sh https://downloads.lede-project.org/snapshots/target
s/ar71xx/generic/lede-ar71xx-generic-tl-wr1043nd-v1-squashfs-factory.bin
```

1) 下载图像文件

```
=====
##### 100.0%
```

2) 下载校验和文件

```
=====
##### 100.0%
```

3) 下载GPG签名

```
=====
##### 100.0%
```

4) 验证GPG签名

```
=====
签名由公钥进行签名, ID为F93525A88B699029, 该系统不存在。
```

在下面提供一个公共密钥服务器url或按回车接受默认建议。按Ctrl-C中止操作。

```
Keyserver使用? [hkp: %% // %% pool.sks-keyservers.net]>
```

```
gpg: 从hkp server pool.sks-keyservers.net请求密钥8B699029
```

```
gpg: key 626471F1: 公钥“LEDE构建系统 (用于无人值守构建作业的LEDE GnuPG密钥)”lede-adm@
lists.infradead.org >>“导入
```

```
gpg: 处理的总数: 1
```

```
gpg: imported: 1 (RSA: 1)
```

```
gpg: Signature made Di 02 Aug 2016 10:10:40 CEST using RSA key ID 8B699029
```

```
gpg: “LEDE构建系统 (LEDE GnuPG密钥, 用于无人值守构建作业)的良好签名”lede-adm@lists.inf
radead.org >>“
```

```
gpg: 警告: 此密钥未通过可信签名认证!
```

```
gpg: 没有迹象表明该签名属于所有者。
```

```
主键指纹: 54CC 7430 7A2C 6DC9 CE61 8269 CD84 BCED 6264 71F1
```

```
子键指纹: 6D92 78A3 3A9A B314 6262 DCEC F935 25A8 8B69 9029
```

5) 验证SHA256校验和

```
=====
lede-ar71xx-generic-tl-wr1043nd-v1-squashfs-factory.bin: 好的
```

验证完成!

=====

固件图像放在`///home/user/lede-ar71xx-generic-tl-wr1043nd-v1-squashfs-factory.bin//`中。

打扫干净。`<用户@主机>: ~$`

## 开发者信息

参与LEDE项目的开发人员需要提供存储在中央keyring.git (<https://git.lede-project.org/?p=keyring.git>)存储库中的*GnuPG*和*usign*公钥。 (<https://git.lede-project.org/?p=keyring.git>)

有关如何生成合适的签名密钥的说明，请参阅密钥生成页面。

## Download.sh

这个便利脚本自动进行密钥生成和验证。

### **Download.sh**

```

#! /usr/bin/env bash
# 执行验证文件下载脚本。
# 退出代码:
# 0 - 文件下载成功并验证
# 1 - 无法下载请求的文件
# 2 - 无法下载sha256sums文件
# 3 - 无法下载sha256sums.gpg文件
# 4 - GnuPG可用但无法验证签名（丢失pubkey，文件完整性错误...）
# 5 - 校验和不匹配
# 6 - 无法将请求的文件复制到其最终目标
# 254 - 脚本被信号中断
# 255 - 适当的下载或校验和实用程序缺失

```

```

[ -n "$1" ] || {
    echo "Usage: $0 <url>" >&2
    exit 1
}

```

```

finish () {
    [-e "/tmp/verify.$$" ] && {
        echo "清理"。
        rm -r "/tmp/verify.$$"
    }
    退出 $1
}

```

陷阱 “完成254” INT TERM

```

destdir = "$ (pwd) "
image_url = "$1"
image_file = "$ {image_url ## * /} "
sha256_url = "$ {image_url%/*} / sha256sums"
pgpsig_url = "$ {image_url%/*} / sha256sums .gpg"
keyserver_url = "hkp: //pool.sks-keyservers.net"

```

```

# 找到一个合适的下载工具
如果 其卷曲 > / dev的/无效; 然后
    下载 () { curl --progress-bar -o "$1" "$2" ; }
elif 其中 wget > / dev / null; 然后
    下载 () { wget -O "$1" "$2" ; }
elif which fetch > / dev / null; 然后
    下载 () { fetch -o "$1" "$2" ; }
else
    echo "找不到合适的下载实用程序，无法下载文件!" >&2
    完成255
fi

```

```

# 找到一个合适的校验实用程序
, 如果 其 sha256sum > / dev的/无效; 那么
    checksum () { sha256sum -c --ignore-missing "sha256sums" ; }
elif 其中 shasum > / dev / null; 那么
    checksum () {
        local sum = "$ (shasum -a 256"$image_file") " ;

```

```

        grep -xF “ $ {sum %% *} * $ image_file ” “sha256sums” ;
    }
else
    echo “没有安装SHA256校验和可执行文件，无法验证校验和!” >&2
    完成255
fi

#检查gpg可用性，
如果 哪个 gpg > / dev / null; 然后
    runpgp () { gpg “$ @” ; }
else
    runpgp () {
        echo “警告：没有GnuPG安装，不能验证数字签名!” >&2
        return 0
    }
fi

mkdir -p “/tmp/verify.$$”
cd “/tmp/verify.$$”

echo “”
echo “1) 下载图像文件”
echo “=====”
下载“ $ image_file ” “ $ image_url ” || {
    echo “无法下载图像文件!” >&2
    完成1
}

echo “”
echo “2) 下载校验和文件”
echo “=====”
下载“sha256sums” “ $ sha256_url ” || {
    echo “无法下载校验和文件!” >&2
    完成2
}

echo “”
echo “3) 下载GPG签名”
echo “=====”
下载“sha256sums .gpg” “ $ gpgsig_url ” || {
    echo “无法下载GPG签名!” >&2
    完成3
}

echo “”
echo “4) 验证GPG签名”
echo “=====”
missing_key = $ ( runpgp --status-fd 1 --with指纹 --verify \
    “sha256sums.gpg” “sha256sums” 2 > / dev的/空| sed的 -ne '! : S ^ * NO_
PUBKEY! p' )

如果 [ -n “ $ missing_key ” ] ; 然后
    回显 “签名由公钥签名，ID $ missing_key ” >&2
    echo “，该系统不存在。” >&2

```

```

    echo ""
>&2

    echo "在下面提供一个公共密钥服务器url或者按Enter键接受" >&2
    echo "默认建议, 按Ctrl-C中止操作。" >&2
    echo ""
>&2

    而 真实的 ; do
        printf "Keyserver to use? [ $ keyserver_url ]>"
        read url; case " $ {url: - $ keyserver_url} " in
            hkp: // * )
                gpg --keyserver " $ {url: - $ keyserver_url} "-
-   recv-keys " $ missing_key " || {
                    echo "无法下载公钥"。 >&2
                    finish 7
                }
                break
            ;;
            * )
                echo "以'hkp: // hostname'形式显示密钥服务器URL。"
>&2
            ;;
        ESAC

    做
网络

    runpgp --with指纹 --verify "sha256sums.gpg" "sha256sums" || {
        echo "无法使用GPG签名验证校验和文件!" >&2
        完成4
    }

    echo ""
    echo "5) 验证SHA256校验和"
    echo "======"
    校验和|| {
        echo "校验和不匹配!" >&2
        完成5
    }


    cp " $ image_file " " $ destdir / $ image_file " || {
        echo "无法写' $ destdir / $ image_file '" >&2
        完成6
    }

    回声 ""
    回声 "验证完成!"
    echo "======"
    echo "固件映像置于' $ destdir / $ image_file '中。"
    回声 ""

    完成0

```

---

 最后修改：2016/12/24 12:04 通过bobafetthotmail

除非另有说明，本维基的内容将根据以下许可证获得许可：**CC Attribution-Share Alike 4.0 International**  
(<http://creativecommons.org/licenses/by-sa/4.0/>)