**Linux内核签名流程**

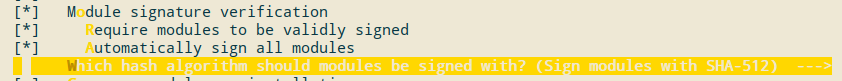
**一：内核的配置**

内核从3.7后开始支持模块签名，这个功能使能以后，内核只允许安装特定key签名的模块；所需内核配置项如下：

CONFIG\_MODULE\_SIG=y：表示开启了签名机制，但是这时候模块签名或不签名都可以使用；

CONFIG\_MODULE\_SIG\_FORCE=y：如果该配置项使能，则模块必须有正确的签名才能正常使用；

CONFIG\_MODULE\_SIG\_ALL=y：内核在编译的时候，并不会主动去给模块签名，除非把该配置项打开；



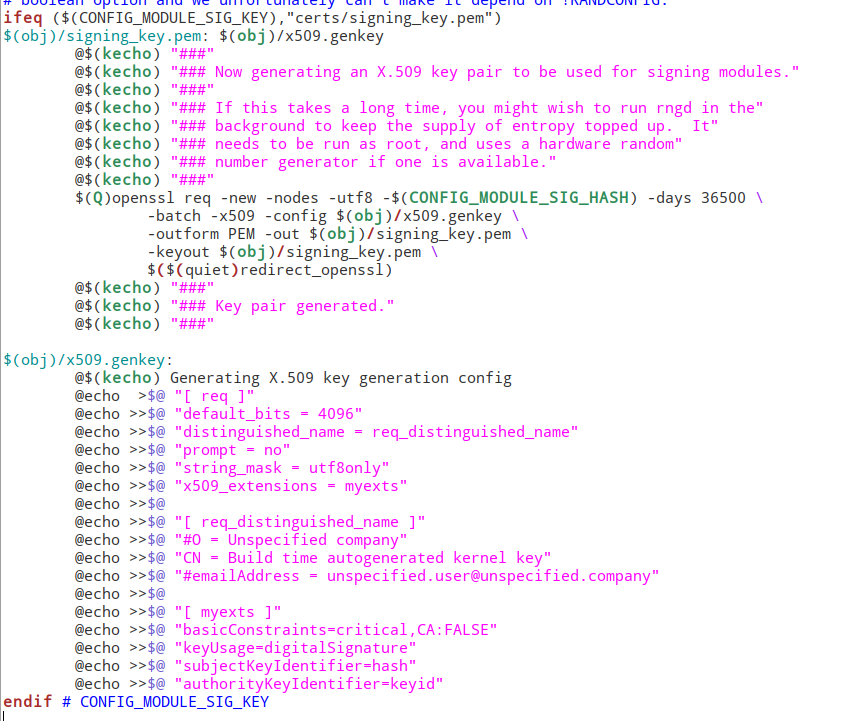
配置CONFIG\_MODULE\_SIG来支持Linux内核中的模块签名验证。它支持两种签名方式:一种相当宽松，另一种是严格的。默认情况下，使用的是许可方法，这意味着Linux内核模块要么必须有有效的签名，要么没有签名。使用严格的方法，必须有有效的签名。在上面的示例中，通过配置CONFIG\_MODULE\_SIG\_FORCE，来使用严格的方法。启用这种严格方法的另一种方法是设置内核引导选项enforcemodulesig=1。在构建Linux内核时，内核模块不会自动签名，除非配置了自动签名所有模块(CONFIG\_MODULE\_SIG\_ALL)。

最后还需要选择用于加密签名的哈希算法。在上面的示例中，采用了SHA-512；

**二：内核签名方式**

在配置了CONFIG\_MODULE\_SIG情况下构建Linux内核，可以选择使用自有的密钥，或者通过配置CONFIG\_MODULE\_SIG\_KEY为“certs/signing\_key.pem”让Linux内核自己构建一组密钥。构建到最后，会在Linux内核源代码的certs/下生成signing\_key.pem和signing\_key.x509。

如果希望使用自己的密钥，可以使用openssl创建密钥对(私钥和公钥)。下面来自certs/Makefile的命令创建了这样一个密钥对。

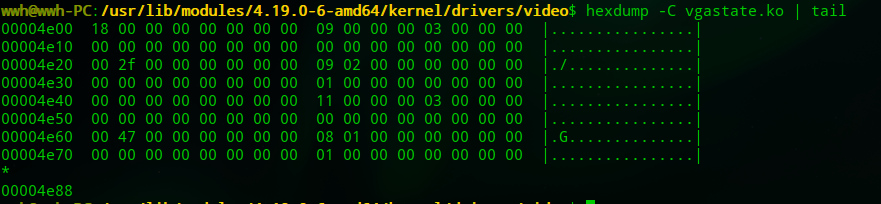


其中，x509.genkey是生成key pair时的配置项，signing\_key.pem signing\_key.x509分别为pem key和数字证书。数字证书会打包进内核，里面有公钥等，用来解密嘛。每编译一次，虽然配置文件每次都相同，但是生成的key pair是不同的;

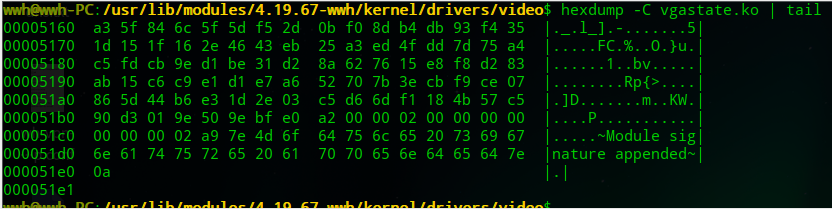
**三：****查看签名信息**

利用下面命令查看设备中的ko文件信息:

hexdump -C vgastate.ko | tail



如上是签名前的数据，签名后的数据如下：



通过如下命令可以移除签名内容：

Sudo strip --strip-debug vgastate.ko

移除签名内容后再尝试加载该驱动就会失败，并会打印：

modprobe: ERROR: could not insert vgastate: Required key not available

**四：管理内核签名模块**

一旦内核启动，并且内核签名模块能正常工作，那么如何能正确处理密钥就很重要了。signing\_key.perm需要被保存在一个安全的位置(除非你要为新的内核创建新的密钥，在这种情况下文件可以被删除)。不能把它放在系统上的/usr/src/linux上，因为恶意软件可以很容易地使用这个密钥来对内核模块(比如rootkit)签名，并进一步危害系统。

**五：手动签署模块**

如果需要手动签名内核模块，可以使用Linux内核中提供的脚本/签名文件脚本。它需要四个参数:

1.哈希算法，如sha512

2.密钥

3.证书(包含公钥)

4.要签名的内核模块