



# CRYPTKEEPER 发现通用密码事件分析报告

安天 CERT

报告初稿完成时间：2017 年 02 月 02 日 15 时 00 分

首次发布时间：2017 年 02 月 02 日 15 时 00 分

本版本更新时间：2017 年 02 月 02 日 15 时 00 分



# 目 录

---

1	事件起因 .....	1
2	事件验证及分析 .....	1
3	事件影响 .....	5
4	关于 CRYPTKEEPER 作者.....	5
5	总结 .....	6
	附录一：参考资料.....	6
	附录二：关于安天.....	6

## 1 事件起因

2017 年 1 月 31 日, Softpedia 网站发布了一篇名为《Cryptkeeper Linux Encryption App Fails at Job, Has One Letter Skeleton Key - "P"》的文章, 其中提及, CryptKeeper 应用在 Debian 9 中存在一个 BUG, 会使得用户为加密文件夹设定的密码被替换为单个字符“p”, 从而使字符“p”作为解密由其加密的文件夹的通用密码。

文章引用了 Kirill Tkhai 于 2017 年 1 月 26 日在 debian 社区 (bugs.debian.org) 提交的 Bug 信息。

安天 CERT 为验证此消息的准确性, 对所涉及模块的源代码进行了分析。

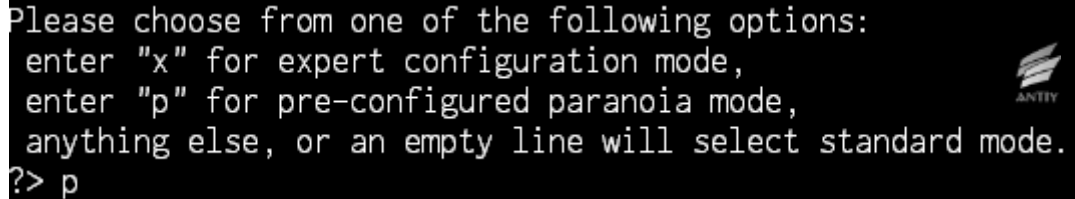
## 2 事件验证及分析

CryptKeeper 是一款工作在 Linux 平台的加密文件夹管理软件, 具有安装、卸载 encfs 文件夹和更改文件夹密码、创建新加密文件夹功能, 可与默认文件管理器集成使用。CryptKeeper 使用 GTK 库编写, 底层对 encfs 模块进行了封装, 使用 AES-192/256 算法对文件进行加密。

产生问题的代码位于 cryptkeeper/src/encfs\_wrapper.cpp, 在该代码中, 作者使用了 encfs 的 -S(Stdinpass) 参数, 从标准输入读取用户输入的密码。

```
127 int encfs_stash_new (const char *crypt_dir, const char *mount_dir, const char *password)
128 {
129     int fd[2];
130
131     assert (pipe (fd) == 0);
132
133     mkdir (crypt_dir, 0700);
134     mkdir (mount_dir, 0700);
135
136     int pid = fork ();
137
138     if (pid == 0) {
139         dup2 (fd[0], 0);
140         // don't want to see encfses stdout bullshit
141         int devnull = open("/dev/null", O_WRONLY);
142         dup2(devnull, 1);
143         close (fd[1]);
144         execlp ("encfs", "encfs", "-s", crypt_dir, mount_dir, NULL);
145         exit (0);
146     }
```

在旧版本的 encfs 中, 会给出两个选项, 让用户手动选择一个加密的模式:



```

Please choose from one of the following options:
enter "x" for expert configuration mode,
enter "p" for pre-configured paranoia mode,
anything else, or an empty line will select standard mode.
?> p
    
```

CryptKeeper 根据旧版本 encfs 的设定，在代码中硬编码了“p\n”选项，来模拟键盘输入，以选择 paranoia 模式(具有更高的加密强度)。以模拟键盘输入的方式实现加密模式选择，显然是不够严谨的，这就为后来的“通用密码事件”埋下了隐患。不过，由于 encfs 并没有提供其它的调用接口，CryptKeeper 也没有更好的选择。

```

148         // paranoid default setup mode
149         //write (fd[1], "y\n", 2);
150         //write (fd[1], "y\n", 2);
151         write (fd[1], "p\n", 2);
152         write (fd[1], password, strlen (password));
153         write (fd[1], "\n", 1);
154         close (fd[1]);
155         int status;
156         waitpid (pid, &status, 0);
157         return !is_mounted(mount_dir);
158     //     return status;
    
```

而本次事件中涉及的 Debian 9，仍处于测试版(unstable)阶段，使用了较新的 encfs(1.9.1-3 版)。新版本的 encfs 在-S 参数的解析过程中不再读取模式，而是使用预配置的标准模式，并直接从输入中读取密码：

```
# root @ kali in ~/Desktop [15:15:38] C:130
$ encfs -S /root/Desktop/test1 /root/Desktop/test2
Creating new encrypted volume.
Standard configuration selected.

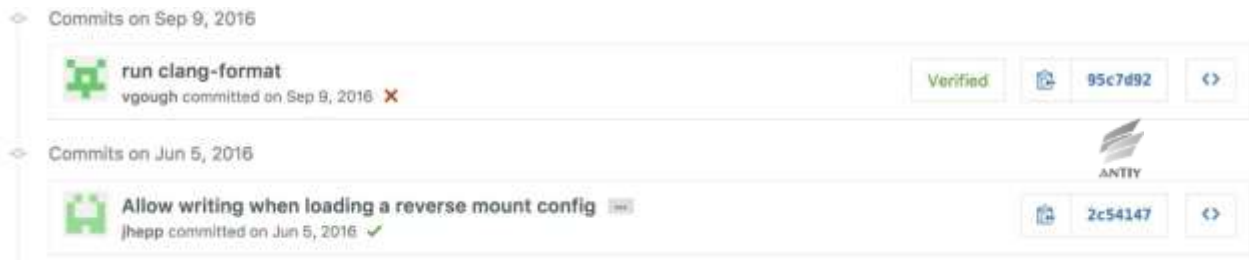
Configuration finished. The filesystem to be created has
the following properties:
Filesystem cipher: "ssl/aes", version 3:0:2
Filename encoding: "nameio/block", version 4:0:2
Key Size: 192 bits
Block Size: 1024 bytes
Each file contains 8 byte header with unique IV data.
Filenames encoded using IV chaining mode.
File holes passed through to ciphertext.

Now you will need to enter a password for your filesystem.
You will need to remember this password, as there is absolutely
no recovery mechanism. However, the password can be changed
later using encfsctl.
```

在源代码中也可以看到对应的流程,其中 useStdin 这个布尔变量在参数为-S 的时候被设置,通过 fgets 读取用户输入的密码。

```
1381 CIPHERKey EncFSConfig::getUserKey(bool useStdin) {
1382     char passBuf[MaxPassBuf];
1383     char *res;
1384
1385     if (useStdin) {
1386         res = fgets(passBuf, sizeof(passBuf), stdin);
1387         // Kill the trailing newline.
1388         if (passBuf[strlen(passBuf) - 1] == '\n')
1389             passBuf[strlen(passBuf) - 1] = '\0';
1390     } else {
1391         // xgroup(common)
1392         res = readpassphrase(_("EncFS Password: "), passBuf, sizeof(passBuf),
1393                             RPP_ECHO_OFF);
1394     }
1395
1396     CIPHERKey userKey;
1397     if (!res)
1398         cerr << _("Zero length password not allowed\n");
1399     else
1400         userKey = makeKey(passBuf, strlen(passBuf));
1401
1402     memset(passBuf, 0, sizeof(passBuf));
1403
1404     return userKey;
1405 }
```

导致 CryptKeeper 出现该问题的 encfs 相关代码,为 2016 年 9 月由 encfs 作者提交。



在 encfs 的新版本中，对于模式的指定如下：

```

951  RootPtr createV6Config(EncFS_Context *ctx,
952                          const std::shared_ptr<EncFS_Opts> &opts) {
953      const std::string rootDir = opts->rootDir;
954      bool enableIdleTracking = opts->idleTracking;
955      bool forceDecode = opts->forceDecode;
956      const std::string passwordProgram = opts->passwordProgram;
957      bool useStdin = opts->useStdin;
958      bool reverseEncryption = opts->reverseEncryption;
959      ConfigMode configMode = (useStdin &&
960                              opts->configMode == Config_Prompt) ? Config_Standard
961                                                                    : opts->configMode;
    
```

而在旧版本中，模式的指定使用了不同的代码：

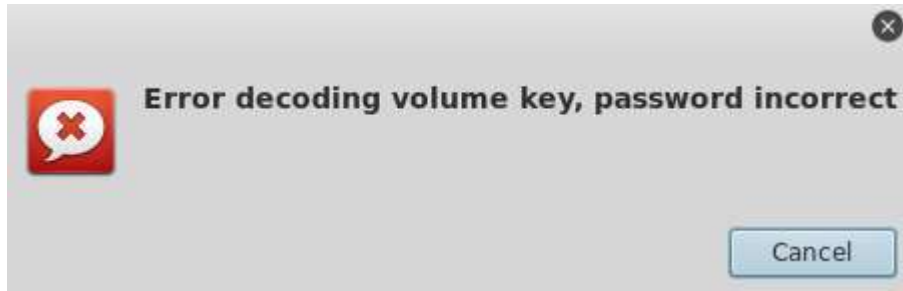
```

950  RootPtr createV6Config(EncFS_Context *ctx,
951                          const std::shared_ptr<EncFS_Opts> &opts) {
952      const std::string rootDir = opts->rootDir;
953      bool enableIdleTracking = opts->idleTracking;
954      bool forceDecode = opts->forceDecode;
955      const std::string passwordProgram = opts->passwordProgram;
956      bool useStdin = opts->useStdin;
957      bool reverseEncryption = opts->reverseEncryption;
958      ConfigMode configMode = opts->configMode;
959      bool annotate = opts->annotate;
    
```

意即，在新旧版本中，对于同一个 Config\_Prompt 模式，是否指定 useStdin(-S)，行为是不一致的。在新版本中，如果指定了-S 并且模式为 Config\_Prompt 的话，会使用标准模式。

因此，CryptKeeper 中硬编码的“p\n”值被 encfs 直接看作密码，而实际应使用的密码被抛弃。由于-S 模式关闭了输入回显，这个 BUG 并不容易发现。

而当用户重新加载时，输入原有密码，就会得到如下的“密码错误”提示：



### 3 事件影响

该事件的影响范围有限，因为：

- 1、CryptKeeper 的使用人数较少，作者甚至一度停止了维护；
- 2、该 Bug 目前只出现在使用了 encfs 最新版本(1.9.1-3)的系统上，而此版本并未被很多发行版所采用；
- 3、在事件发生后，CryptKeeper 作者已从 Debian 9 的官方源中移除了自己的软件，并将在修复后重新上传该软件。

不过，对于 CryptKeeper 的用户来说，该事件的影响却是恶性的。一方面，不知情的用户在解除挂载后，将无法再次访问自己的加密文档；另一方面，用户数据在攻击者面前毫无加密强度可言，隐私数据可以被轻松解密获取。

### 4 关于 CryptKeeper 作者

CryptKeeper 的作者 Tom Morton 预留邮箱为 t-morton@blueyonder.co.uk，而 blueyonder.co.uk 域名的注册单位为维珍媒体 (Virgin Media)。目前，blueyonder.co.uk 和 virginmedia.com 均指向同一页面。Virgin Media 是一家英国公司，为企业和消费者提供固定和移动电话、电视、宽带互联网服务。

作者于 1995 年在英国布里斯托大学 (University of Bristol) 获得政治学学士学位，专研研究美国政治、国际关系、政治理论和撒切尔主义。作者在维珍媒体担任执行规划总监 (Executive Planning Director) 期间，在 github 启动了 CryptKeeper 项目，代码最早提交时间为 2007 年 7 月 12 日。目前作者在美国 R/GA 公司任高级副总裁 (SVP Strategy)。



Tom Morton



<https://www.linkedin.com/in/realtomorton/zh-cn>

#### Tom Morton

SVP, Strategy at R/GA

New York, New York | 市场营销与广告

目前就职 R/GA, Strategist-at-Large

曾经就职 co:collective, Goodby Silverstein & Partners, Havas

教育经历 University of Bristol

建立联系

向Tom发送 InMail

## 5 总结

---

通过上述分析可以看出,这个自 2007 年起就一直以硬编码形式存在于 CryptKeeper 源代码中字符“p”,之所以在 2016 年成为可以破解加密数据的“通用密码”,是因为其调用的 encfs 修改了一个参数(-S)的执行逻辑。

由于此事件给用户造成的后果是明显可感知的——用户使用所设定密码无法解密数据——我们可以初步定性这是一个 Bug,而暂不倾向于认为这是一个由开发者(包括 CryptKeeper 的开发者 and encfs 的开发者)预制的后门,也暂时没有迹象表明这是一次由攻击者入侵开发环境造成的代码污染。

这个案例再度说明了系统安全的复杂性——特别是系统安全和数据安全“连接部”的脆弱性。对于围绕开源系统(也包括闭源软件)构建的开发环境的环境安全、过程安全,以及更广泛的供应链安全,我们还需擦亮双眼。

## 附录一：参考资料

---

[1] 事件原始新闻

<http://news.softpedia.com/news/cryptkeeper-linux-encryption-app-fails-at-job-has-one-letter-skeleton-key-p-512432.shtml>

[2] 新闻消息来源

<https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=852751>

[3] Wikipedia 的 Virgin Media 词条

[https://en.wikipedia.org/wiki/Virgin\\_Media](https://en.wikipedia.org/wiki/Virgin_Media)

[4] 作者的 linkedin 页面:

<http://www.linkedin.com/in/realtommorton>

[5] CryptKeeper 源代码

<https://github.com/tomm/cryptkeeper>

[6] encfs 源代码

<https://github.com/vgough/encfs>

## 附录二：关于安天

---

安天从反病毒引擎研发团队起步,目前已发展成为以安天实验室为总部,以企业安全公司、移动安全公司为两翼的集团化安全企业。安天始终坚持以安全保障用户价值为企业信仰,崇尚自主研发创新,在安



全检测引擎、移动安全、网络协议分析还原、动态分析、终端防护、虚拟化安全等方面形成了全能力链布局。安天的监控预警能力覆盖全国、产品与服务辐射多个国家。安天将大数据分析、安全可视化等方面的技术与产品体系有效结合，以海量样本自动化分析平台延展工程师团队作业能力、缩短产品响应周期。结合多年积累的海量安全威胁知识库，综合应用大数据分析、安全可视化等方面经验，推出了应对高级持续性威胁（APT）和面向大规模网络与关键基础设施的态势感知与监控预警解决方案。

全球超过三十家以上的著名安全厂商、IT 厂商选择安天作为检测能力合作伙伴，安天的反病毒引擎得以为全球近十万台网络设备和网络安全设备、近两亿部手机提供安全防护。安天移动检测引擎是全球首个获得 AV-TEST 年度奖项的中国产品。

安天技术实力得到行业管理机构、客户和伙伴的认可，安天已连续四届蝉联国家级安全应急支撑单位资质，亦是中国国家信息安全漏洞库六家首批一级支撑单位之一。安天是中国应急响应体系中重要的企业节点，在红色代码 II、口令蠕虫、震网、破壳、沙虫、方程式等重大安全事件中，安天提供了先发预警、深度分析或系统的解决方案。

<http://www.antiy.com>（中文）

关于反病毒引擎更多信息请访问：

<http://www.antiy.net>（英文）

关于安天反 APT 相关产品更多信息请访问：

<http://www.antiy.cn>