

badkeys



Finding broken cryptographic keys

<https://badkeys.info/>

Hanno Böck



Security

GitHub security update: revoking weakly-generated SSH keys

On September 28, 2021, we received notice from the developer Axosoft regarding a vulnerability in a dependency of their popular git GUI client - GitKraken. An underlying issue with a dependency, called `keypair`, resulted in the GitKraken client generating weak SSH keys.

"There is no haveibeenpwned for public keys as far as I know"

user [jornane](#) on [lobste.rs](#), 10/2021

Checking cryptographic public keys for known vulnerabilities

Enter Key:

```
-----BEGIN RSA PUBLIC KEY-----  
MIIBCgKCAQEAwJZTDExKND/DiP+LbhTIi2F0hZZt0PdX897LLwPf3+b1G0CUj10H  
BZvVqhJPJtOPE53W68I0NgVhaJdY6bFOA/cUUIFnN0y/Z0J0JsPNle1aXQTjxAS+  
FXu4CQ6a2pzcU+9+gGwed7XxAKIVCiTprfmRCI2vIKdb61S8kf5D3YdVRH/Tq977  
nxyYeosEGYJFB0IT+N0mqca37S8hA9hCJyD3p0AM40dD5M5ARAxpAT7+oq0XkPzf  
zLtCTaHYJK3+WAce121Br4NuQJPqYpVxniUPohT4YxFTqB7vwX2C4/gZ2ldpHtlg  
JVAHT96n0snLz+EPa5GtwxtALD43Cw0lWQIDAQAB  
-----END RSA PUBLIC KEY-----
```

OK

CLEAR

Supported are X.509 certificates (CRT), Certificate Signing Requests (CSR), PEM public and private keys according to PKCS #1 and PKCS #8, and SSH public keys. (While supported, uploading private keys is obviously discouraged for production keys.)

Fill with test data

NORMAL RSA KEY

ROCA

FERMAT

DEBIAN OPENSLL

MANY ZEROS

CORRUPT

UNUSUAL SIZE

SMALL KEY

EXPONENT 3

ED25519 RFC EXAMPLE

badkeys

A website, tool and library to check cryptographic keys for known vulnerabilities

Key Generation Vulnerabilities

- Shared prime factors
- Return of Coopersmith's attack / ROCA
- Fermat attack
- Debian OpenSSL Bug
- keypair / Gitkraken bug
- "Public Private Keys"

Debian OpenSSL Bug (CVE-2008-0166)

```
-----  
Debian Security Advisory DSA-1571-1                security@debian.org  
http://www.debian.org/security/                   Florian Weimer  
May 13, 2008                                     http://www.debian.org/security/faq  
-----
```

```
Package      : openssl  
Vulnerability : predictable random number generator  
Problem type : remote  
Debian-specific: yes  
CVE Id(s)    : CVE-2008-0166
```

Luciano Bello discovered that the random number generator in Debian's openssl package is predictable. This is caused by an incorrect Debian-specific change to the openssl package (CVE-2008-0166). As a result, cryptographic key material may be guessable.

Keys depended on a limited number of factors like the PID and the architecture, limiting the number of possible keys to a few ten thousand

Old bugs never die



Matt Palmer

Mar 7, 2020, 3:48:48 AM

to mozilla-dev-s...@lists.mozilla.org

(Pre) Certificate <https://crt.sh/?id=2531502044> has been issued with a known weak key, specifically Debian weak key 2048/i386/rnd/pid17691. I believe this issuance to be in contravention of SSL.com's CPS, version 1.8, section 6.1.1.2, which states "SSL.com shall reject a certificate request if the request has a known weak Private Key".

- Matt

[Matt Palmer on mozilla-dev-security-policy, 2020](#)

Detecting the Debian OpenSSL bug

Existing tools and lists of affected keys were not exactly great

- Some of the old tools no longer worked on modern systems
- All collections of affected keys were incomplete
- Information about the exact details of the bug was confusing, incomplete, and sometimes wrong

Debian OpenSSL Bug variations

- PID (0 to 32767)
- OpenSSL and OpenSSH
- Different output if .rnd file exists
- Older and newer OpenSSL versions differ if the .rnd file does not exist
- Architectures: 32/64 bit, x86 vs. ppc/others vs. mips
- Key size
- RSA, DSA, Elliptic Curves (!)

<https://github.com/badkeys/debianopenssl/>

Earlier this year

"I should test DKIM keys with badkeys"

DKIM

TXT record at *key1._domainkey.hboeck.de*:

v=DKIM1; k=rsa; p=MIIBIjANBgkqhkiG9w0BAQE[...]

E-Mail header:

DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=hboeck.de; s=key1; t=1715197611;

bh=Z9fPSuWvmaUL/fgn9g0k2ORYPJe3Y3Vc5NiKvQJXc2w=; h>Date:From:To:Subject:Message-ID:MIME-Version:Content-Type: Content-Transfer-Encoding; b=TNyZHQd[...]

How to scan DKIM

Get lots of e-mails and extract selector/domain combinations

How to scan DKIM (better)

Try common selectors like dkim, mail, etc., with top domains

Scanning Tranco 1 Top Million list

Around 350,000 TXT records with a valid RSA key.

855 vulnerable to Debian OpenSSL bug (0.24%).

Domains with vulnerable keys

**@cisco.com, @oracle.com, @skype.net, @github.partners,
@partner.crowdstrike.com, @partners.dropbox.com, @1password.com, @seznam.cz**

Why?

- 2006: Debian OpenSSL bug was introduced
- 2007: DKIM was published (RFC 4870)
- 2008: Debian OpenSSL bug was found

**Most affected keys were configured as a CNAME
to a host belonging to the company Cakemail**

Trying to disclose a security issue to security@cakemail.com

We're writing to let you know that the group you tried to contact (security) may not exist, or you may not have permission to post messages to the group.

There were these logos...

Inbox

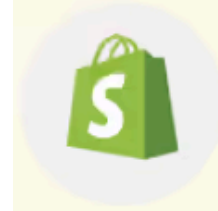


admin

5 Apr

BIMI is nonsense

This should show the BIMI logo from Ent...



admin

5 Apr

Give me your password

Please send your Shopify username and...



admin

5 Apr

Give me your password

Please send your CrowdStrike username...



admin

5 Apr

Give me your password

Please send your Dropbox username an...



More on DKIM findings and BIMl: Talk at MiniDebConf

<https://16years.secvuln.info/>

Fermat Attack



RSA

$$**N = p * q**$$

If you can calculate p, q from N, you can break RSA (factoring)

Fermat Factorization (1643)

Simple algorithm that can efficiently find prime factors if they are of similar size

How to not generate RSA keys

- Generate random number x
- Find next prime after x and use as p
- Find next prime after p and use as q

Are there such RSA keys?

Printers from Canon and Fujifilm generated keys breakable with Fermat Factorization (Safezone library from Rambus, CVE-2022-26320)

<https://fermatattack.secvuln.info/>

Public Private Keys

The screenshot shows a GitHub repository page for 'openssl / openssl'. The repository is public and has 10.1k forks and 25.5k stars. The current view is for the file 'x509-check-key.pem' in the 'certs' directory. A commit by 'InfoHunter and mattcaswell' is highlighted, dated 6d2523e · 7 years ago. The commit message is 'Add test cases for X509_check_private_key'. The file content is displayed as follows:

```
1  -----BEGIN PRIVATE KEY-----
2  MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBywggSiAgEAAoIBAQCd6jpgFiM/ZW6d
3  CJLEIxMkK7rH7MRL93w32o5duTwtT1cs/y+yLfey0l5tYBzGMxjUPNeYGTBqiuZ
4  6ueVyMvbe3wymXPP+zzoaq3if3Jycb+1gurSyiQpF6T1PLmfJDgQQT0XnI7qRwHI
5  5FJTvKM9mpv3iKohBseT/a8yfdk27zFYrSMZjfaqZc+0a18bHi/SgNN36Lj+vnPc
6  s2DzS8ymBJ10Zq6icy6xL30sHDKPOKKrD8+EJ6suUm5CpLL4N6jP0mk9Dj7XQv2Y
7  woX2S0Ys6dFpHuGBJ1NngBW/@Zm9oseD0xxqpLPGIYa8nN7BIRTwAJEhkmKTEi9P
8  8APIi6DVAgMBAACggEAMWkKnuo0WVXJiIUaP8GjykJzHP8uZH6paxa4zAYxmEd9
9  TbZbj08PE30UHmr2KA1IVoMLwynyHM68Ie2MTmepUaGPuN1e8YVV83vpsIckLj79
10 NzQheZcaPwLSihFYGz1f9WYUUYEBDrjtDAi04dKSWUI5LvIqEu9mHx4vZWMPRIqP
11 mrtp3CH34ViJL4v4TtvEeu0vLf4mYpfWe1l7U2eYSqc00lCwk7nd/JCzpPWA7C
12 TQZSTtp5AQ40T7LPFZIGs/87Qi8fuEEvN+6rt07r0j6/gPOVa2xoj4a7MJYsxi90
13 s1xA8Q+xjUEnjHth1MLCrmHYbJuwptIqgPTkVvB20QKBgQDSAywBvs7PDdt+BLTc
14 6J4g/g0L/17ATysmhUGJ6VxrNuLViLtiFeyf3p4vj/fSa2y4ZnP/hHovzfces1Bd
15 6YXtPGIuRn0nVdlYx2Y/0Grw0baxRAIW8D6Z4ms1n8hesGsssteKZeaT4ojIPpJS1
16 c1UtextX50BLYaiFwTb1Q6bAwKBgQDAfpbrlBN4936glc5uFmKNvFfNB8P30+Bk
17 DFtth5TMsCL406aUlI14lkBrXAgUTndRai2cwYD9ffsXQmm+yx1q5K06akeAaueq
18 WMo3ViZnxK8Fe4oF4M90oaEQRcVmV5jFMKH9S268B8/x96lNh/i7M58nB5AeNDlV
19 AMyHW2vhRwKBgAxduXKk3KKei0Uhw9ECNYV1z5mnmMD9tlz1Uik5mQky7BLV96
20 MQ085Q2h6ZLPVoijJ91s3JECMDIXBu1wub0daB6XW0sqh/DNVPz2An4Jqzt660SW
21 4ujGx09SCEdjFfx8/UnS0t+VFW0MamFA2EwkSpjjVj26E2VfMckMA58nAoGADabs
22 vTh7SREEgg8d30DpjHPXJktuspzsRSw7L8F15C55zHv2TINcXJkLaJHwYnpPzA5j
23 vbr7Uv8kv7n2FfoB1BsQop/3AjySwZoafWI2xxVD9HewimQvT7xw1/iaz29w/mU8
24 l+JJsdw9m0dVkpWcbBvkS0QI5RAnK650r/BHvECgYB6s9Qp5os0CdtPli7MYyD6
25 mw+61DSgThUgKa7j96NG2ToYeNwTdf2Fd4Xa7s6MwryaGY+IMSRga24CM+WvaaAL
26 iGZLY8dfpM/yDr0pva4WF66ARajDhNx1wv0BQJpHnldX0G4gYcIsIWgUhz04eH8
27 370zKradFq+avGmtCBeV8A==
28 -----END PRIVATE KEY-----
```

Many Public Private Keys

- Testcases in software
- Examples in documentation
- Hardcoded keys in software or firmware
- Leaks
- ...

**Any recommendations how to deal with this?
(Github has no working security contact)**

Plans for the Future of badkeys

Thanks to funding by NLnet/NGI0



Increase coverage of Public Private Keys

<https://github.com/badkeys/keyfinder/>

Monitoring

WebPKI, DNSSEC, DKIM

Key Compromise Service

**You submit a compromised key, badkeys takes care of it
(Certificate Revocation, added to blacklist)**

Call for help

Do you have any private keys you want to share with me?

Thanks for listening
Please use badkeys!



Questions?

<https://badkeys.info>