

Google Reveals Cryptographic Secrets

Emin Islam Tatlı

Department of Computer Science, University of Mannheim

tatli@th.informatik.uni-mannheim.de

<http://th.informatik.uni-mannheim.de/people/tatli.shtml>

Google hacking is a term to describe the search queries that find out security and privacy flaws. Finding vulnerable servers and web applications, server fingerprinting, accessing to admin and user login pages and revealing username-passwords are all possible in Google with a single click. Google can also reveal secrets of cryptography applications, i.e., *clear text and hashed passwords, secret and private keys, encrypted messages, signed messages* etc. In this paper, advanced search techniques in Google and the search queries that reveal cryptographic secrets are explained with examples in details.

1 Motivation

Having an index with over 25 billion entries, Google is the most popular web search engine. It indexes any information from web servers thanks to its hardworking web crawlers. But many sensitive data that should be kept secret and confidential are indexed by Google, too. Vulnerable servers and web applications, username-passwords for login sites, admin interfaces of database servers and online devices like web cameras without any access control, reports of security scanners and many more private information are available to hackers via Google.

This paper focuses on the advanced search queries that enable users to search different cryptographic values which are expected to stay private and safe. The paper is organized as follows: Section 2 summarizes the useful parameters for the advanced search in Google. In Section 3, examples of search queries for each type of cryptographic secret are illustrated. Finally, Section 4 explains possible security measures against Google hacking.

2 Advanced Parameters

Google supports many parameters for the advanced search and filters its results according to the parameters given by the user.

The *[all]inurl* parameter is used to filter out the results according to if the url contains a certain keyword or not. If more keywords are needed, the *allinurl* parameter should be used. *[all]intitle* filters the results according to the title of web pages. *[all]intext* searches keywords in the body of web pages. With the parameter *site* you can do host-specific search. *filetype* and *ext* parameters have the same functionality and are needed to filter out the results based on the file extensions like html, php, asp etc. The minus sign (-) can be put before any advanced parameter and reverses its behavior. As an example, a search containing the parameter *-site:www.example.com* will not list the results from www.example.com. The sign "|" stands for the logical OR operation.

3 Google Search for Cryptographic Values

From the cryptographic perspective, Google reveals also cryptographic secrets. Google can find out hashed passwords, secret keys, public and private keys, encrypted and signed files. What you need to do is only to enter the relevant search terms as explained in the following sections and click the search button.

3.1 Hashed Passwords

Database structures and contents can be backed up in *dump* files. The following query searches for SQL clauses that may contain usernames and passwords in cleartext or in hashed values within dump files. Hash and encryption relevant keywords can also be searched within files.

```
"create table" "insert into" "pass|passwd|password"(ext:sql | | ext:dump | ext:dmp)
intext:"password|pass|passwd" intext:"md5|sha1|crypt" (ext:sql | ext:dump | ext:dmp)
```

3.2 Secret Keys

Since the secret keys are generated mostly as session keys and destroyed after the session is closed, they are not stored on disks permanently. But there are still some applications that need to store secret keys, e.g., Kerberos [9] shares a secret key with each registered principal for authentication purposes.

The following query lists the configuration files of a key distribution center (KDC) in Kerberos. Within the configuration files, the path of principal databases which contain principal ids and their secret keys is specified.

```
inurl:"kdc.conf" ext:conf
```

To find dumped Kerberos principal databases:

```
inurl:"slave_datatrans" OR inurl:"from_master"
```

Java provides a tool named *keytool* to create and manage secret keys in keystores. The extension of such keystores is *ks*. The following query searches for java keystores that may contain secret keys. Note that keytool can also manage private keys and certificate chains.

```
keystore ext:ks
```

3.3 Public Keys

Public keys, as the name implies, are public information and not secret. But for the sake of completeness, the search queries that list public keys are also written in this section.

To list PGP public key files:

```
"BEGIN PGP PUBLIC KEY BLOCK" (ext:txt | ext:asc | ext:key)
```

To list public keys in certificate files:

```
"Certificate:Data:Version" "BEGIN CERTIFICATE" (ext:crt | ext:asc | ext:txt)
```

3.4 Private Keys

Private keys should be kept *secret* for personal use but the following search queries show that people do not care about it and make it publicly accessible.

```
"BEGIN (DSA|RSA)" ext:key
```

```
"BEGIN PGP PRIVATE KEY BLOCK" inurl:txt|asc
```

Gnupg [5] encodes the private key in *secring.gpg*. The following search reveals *secring.gpg* files:

```
"index of" "secring.gpg"
```

3.5 Encrypted Files

For confidentiality, cryptography provides encryption of data. By encrypting, one can store sensitive files and emails securely on local storage devices. The following queries search for encrypted files and emails. It is sure that you need to know the relevant keys to decrypt but as shown in the previous examples, it is also possible to find secret keys and private keys. Besides, other crypto analysis techniques can help to decrypt the encrypted files.

The files that are encrypted with Gnupg get the extension *gpg* for binary encoding and the extension *asc* for ASCII encoding. The following first query searches files with *gpg* extension and tries to eliminate signed and public key files from the results. The second query lists ASCII encoded encrypted files. But note that signed files have also the same pattern and can be returned with the second query:

```
-"public|pubring|pubkey|signature|pgp|and|or|release" ext:gpg
```

```
-"BEGIN PGP MESSAGE" ext:asc
```

Many encryption applications use the extension *enc* for the encrypted files. There are some exceptions like AxCrypt File Encryption Software [6] which uses the extension *axx* for encrypted files:

```
-intext:"and" (ext:enc | ext:axx)
```

In XML Security, the encrypted parts of messages are encoded under *CipherValue* element:

```
"ciphervalue" ext:xml
```

3.6 Signed Messages

Digital signatures provide integrity, authenticity and non-repudiation in cryptography. The following searches list some signed messages, signed emails and file signatures.

To list pgp signed messages (*emails excluded*):

```
"BEGIN PGP SIGNED MESSAGE" -"From" (ext:txt | ext:asc | ext:xml)
```

To list signed emails:

```
"BEGIN PGP SIGNED MESSAGE" "From" "Date" "Subject" (ext:eml | ext:txt | ext:asc)
```

To list file signatures:

```
-"and|or" "BEGIN PGP SIGNATURE" ext:asc
```

4 Countermeasures

Google hacking can be very harmful and therefore the required security measures should be taken against it. One method is using automatic scan tools [2, 3, 4] that search possible Google hacks for a given host. You can use the tools to search for the available flaws and risks in your system. The tools mostly use the hack database [1] when they do scan. Another solution is integration of robots.txt (robots exclusion standard) [7] files in your system. Web crawlers (*hopefully*) respect the directives specified in robots.txt. Providing this, you can prevent the crawlers from indexing your sensitive files and directories. The last and the most advanced suggestion is installing and managing Google honeypots [8] in your system and trying to figure out the behaviour of attackers before they deal with your *real* system.

References

- [1] Google Hacking Database. <http://johnny.ihackstuff.com/index.php?module=prodreviews>.
- [2] GooLink- Google Hacking Scanner. <http://www.ghacks.net/2005/11/23/goolink-scanner-beta-preview/>.
- [3] SiteDigger v2.0 - Information Gathering Tool. <http://www.foundstone.com>.
- [4] Johnny Long. Gooscan: Google Security Scanner. <http://johnny.ihackstuff.com/modules.php?op=modload&name=Downloads&file=index&req=getit&lid=33>.
- [5] The GNU Privacy Guard. [http://www.gnupg.org/\(en\)/index.html](http://www.gnupg.org/(en)/index.html)
- [6] AxCrypt File Encryption Software for Windows. <http://axcrypt.axantum.com>
- [7] Robots Exclusion Standard. <http://en.wikipedia.org/wiki/Robots.txt>
- [8] Google Hack Honeypot Project. <http://ghh.sourceforge.net>
- [9] Kerberos:The Network Authentication Protocol. <http://web.mit.edu/kerberos/>