GitHub - C-Sto/gosecretsdump: Dump ntds.dit really fast

By C-Sto

Archived: 2026-04-05 21:21:19 UTC

Have you been using Impacket to dump hashes out of (large) NTDS.dit files, and become increasingly frustrated at how long it takes? I sure have!

All credit for the original code to the impacket devs, it's much more complicated than I anticipated.

This is a conversion of the impacket secretsdump module into golang. It's not very good, but it is quite fast. Please let me know if you find bugs, I'll try and fix where I can - bonus points if you can provide sample .dit files for me to bash against.

Features

- Dumps dits very fast. Operations that usually take hours are now done in minutes.
- Can dump SAM/SYSTEM backups
- Can dump local SAM/SYSTEM (must be run as the machine account/SYSTEM)
- A somewhat usable interface for integration other other tooling (See lib example below)

Usage

You will need to obtain the NTDS.dit and SYSTEM file from the target domain controller as normal. This won't dump anything remotely, just local (for now at least).

```
 -enabled
      Only output enabled accounts
-history
      Include Password History
-livesam
      Get hashes from live system. Only works on local machine hashes (SAM), only works on Windows.
-noprint
      Don't print output to screen (probably use this with the -out flag)
-ntds string
      Location of the NTDS file (required)
-out string
      Location to export output
-sam string
      Location of SAM registry hive
-status
      Include status in hash output
-stream
```

```
        Stream to files rather than writing in a block. Can be much slower.
  -system string
        Location of the SYSTEM file (required)
  -version
        Print version and exit
```

Example (there is a test .dit and system file in this repo)

```
gosecretsdump -ntds test/ntds.dit -system test/system
```

Comparison

Using a large-ish .dit file (approx 1gb)

Impacket secretsdump.py

```
time ./secretsdump.py local -system ~/go/src/github.com/c-sto/gosecretsdump/test/big/registry/SYSTEM
<snip>
./secretsdump.py -system registry/SYSTEM -ntds  local  1197.36s user 12.01s system 98% cpu 20:23.78
```

gosecretsdump

```
time go run main.go -system ~/go/src/github.com/c-sto/gosecretsdump/test/big/registry/SYSTEM -ntds ~
<snip>
go run main.go -system  -ntds  26.28s user 3.78s system 114% cpu 26.178 total
```

Lib

So you want to use this in your cool Go implant? that should be easy. The pattern for all the 'dumping' functions is as follows:

note It's likely that the api will undergo changes. I'll try to keep to semver, but please understand that I don't really have any idea what I'm doing.

```
//Create the reader flavour of your choice
dr, err = samreader.New("C:\\pentest\\system.hive", "C:\\pentest\\sam.hive")
if err != nil {
      return err
}

//Get the output channel
dataChan := dr.GetOutChan()

//start dumping
go dr.Dump()
```

```
//read from the output channel (the channel will be closed once dumping is complete)
wg := sync.WaitGroup{}
wg.Add(1)
go func(){
      defer wg.Done() //This probably won't actually work, I can never remember if defer works on in
      for dh := range dataChan{
            fmt.Println("%+v\n", dh)
      }
}()
//do other things while you wait
wg.Wait()
```

Source: https://github.com/C-Sto/gosecretsdump