





[Product](#) [Solutions](#) [Resources](#) [Open Source](#) [Enterprise](#) [Pricing](#)
[Sign in](#) [Sign up](#)


 Jumpsec Labs / TokenSmith Public


 Notifications


 Fork 34


 Star 258


 Code


 Issues

 Pull requests

 Actions

 Projects

 Security

 Insights

main

Code

**gladstomych-jumpsec**
Update version tag
5aebff1 · 3 months ago
🕒 8 Commits

|              |   |              |
|--------------|---|--------------|
| 📁 cmd        | Update version tag                        | 3 months ago |
| 📁 internal   | Update version tag                        | 3 months ago |
| 📁 media      | Add banner logo                           | 3 months ago |
| 📁 release    | Golang formatting changes                 | 3 months ago |
| 📄 .gitignore | Init                                      | 3 months ago |
| 📄 LICENSE    | Init                                      | 3 months ago |
| 📄 README.md  | Update README.md - add default flag...    | 3 months ago |
| 📄 build.sh   | Add build script & fix default User Agent | 3 months ago |
| 📄 go.mod     | Init                                      | 3 months ago |
| 📄 go.sum     | Init                                      | 3 months ago |
| 📄 main.go    | Golang formatting changes                 | 3 months ago |

### About

TokenSmith generates Entra ID access & refresh tokens on offensive engagements. It is suitable for both covert adversary simulations and penetration tests with the tokens generated working out of the box with many popular Azure post exploitation tools.

- 📖 Readme
- 📄 GPL-3.0 license
- 📈 Activity
- 📋 Custom properties
- ★ 258 stars
- 👁 4 watching
- 🔗 34 forks


Report repository

---

### Releases

🏷 1 tags

# TokenSmith



TokenSmith generates Entra ID access & refresh tokens on offensive engagements. Built with OpSec in mind it is suitable for both covert adversary simulations, penetration tests or sysadmin tasks. The tokens generated works out of the box with many popular Azure offensive tools.

**TL;DR** - Run `./tokensmith authcode` and authenticate using the generated URL in a browser, then paste the relevant redirected URI with the `code` parameter back to the CLI tool, and it will redeem the usable tokens for you.

**Intune Bypass** - This release of TokenSmith (v0.8) supports bypassing Intune compliant device Conditional Access, i.e. you could log in from a non-compliant device to get access tokens using TokenSmith even when it is explicitly required - just add the `-i / --intune-bypass` flag to `authcode` !

### Languages

Go 99.0%

Shell 1.0%

## Installation

### Build From Source

```
git clone https://github.com/jumpseclabs/tokensmith.git
cd tokensmith
go get .
go build -o tokensmith main.go

# to build for Windows
GOOS=windows go build -o tokensmith.exe main.go
```

### Use a Release

Pre built x64 binaries are in the `release` directory.

## Usage

### Getting Tokens

The default combination of client ID & resource should work out of the box in a wide variety of cases. There are additional flags to customise the redirect URI, scope & User-agent string used for OpSec considerations or specific use cases.

#### Authcode Flow

**Step 1** - Run tokensmith with `authcode` .

```
./tokensmith authcode [-c client_id] [-r resource] [-R redirect_uri]
```

explainer\_default.mp4 ▾

#### Authcode Flow with Intune Bypass

Additionally, if you need to bypassing Intune Compliant device Conditional Access, add the `--intune-bypass` flag (or simply `-i` ):

```
./tokensmith authcode --intune-bypass [optional flags]
```

explainer\_intune.mp4 ▾

**Step 2** - Authenticate on a web browser using the link TokenSmith generates.

**Step 3** - Paste the redirecting URI after authenticating into the tokensmith. Refer to the [companion blog post at JUMPSEC labs](#) for more details), and press RETURN. If all goes well, you should see:

```
...
[+] SUCCESSFULLY REDEEMED TOKENS!

[+] Access Token:
eyJ...

[+] Refresh Token:
1.A...
```

Refresh Token Flow

Getting new tokens from refresh tokens

```
./tokensmith reftoken -r REFRESH_TOKEN [-c client_id] [-r resource]
```

Integration with open source offensive tooling

Assuming you have obtained tokens successfully, you can use them with GraphRunner like so:

```
Import-Module .\GraphRunner.ps1

$Accesstoken = eyJ...
$Refreshtoken = 1.A...
$tenantID = <uuid>

Invoke-ImportTokens -AccessToken $Accesstoken -RefreshToken $RefreshToken -TenantID $tenantID
Invoke-CheckAccess -Tokens $tokens
```

Usage with Roadrecon

```
# if you used intune-bypass
roadrecon auth --refresh-token 1.A... -c 9ba1a5c7-f17a-4de9-a1f1-617f-4b8d-4d7d-4b8d-4d7d

# if you used default flags
roadrecon auth --refresh-token 1.A... -c 1fec8e78-bce4-4aaf-ab1b-545f-4b8d-4d7d-4b8d-4d7d
```

Design Considerations

Arguably one of the loudest thing an attacker can do in Entra ID is to authenticate, therefore TokenSmith's core mode `authcode` is designed with a lot of flexibility in the

Users are free to choose to authenticate on whichever browser they fancy, using either password/MFA, importing ESTSAUTHPERSISTENT cookies, or simply using an active browser session to satisfy the authentication grant. TokenSmith does not come bundled with a web browser and does not even need to be run on the same host as the browser.

**Note that:** The interactive authentication can stay on the beachhead device and the operator does not need to worry about running a BOF or a foreign binary to risk detection. They only need to copy URLs to and from the endpoint.

Features and Roadmap

Currently Supported Flows:

- Authorization Code flow

- Refresh Token flow
- Intune Compliant Device Bypass

Planned Features:

- More Detailed Documentation & Usage Wiki
- Device Code Flow
- Check Access
- Slimmed Down PowerShell version

## License

This project is licensed under GPLv3.

Maintained by: [Sunny Chau @ gladstomych](#)