Nim for Adversarial Operations

Getting hands-on with NimPlant C2

Cas van Cooten

2022-03-09

#AdversaryGuru

00 | About

```
[cas@adversaryvillage ~]$ whoami
```

- Offensive Security Enthusiast, Red Team Operator, and hobbyist Malware Developer
- Likes building malware in Nim
- Authored offensive tools such as Nimpackt, and more recently NimPlant
- Semi-pro shitposter on Twitter

Cas van Cooten

casvancooten.com

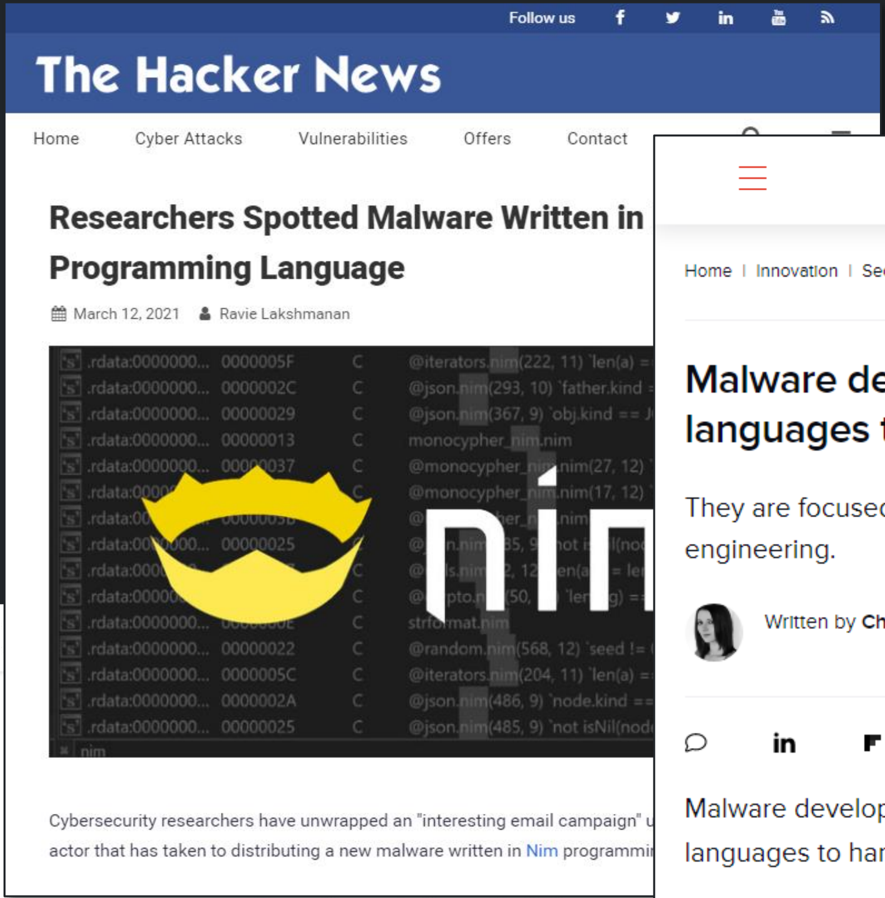
@chvancooten

chvancooten

/in/chvancooten

01 | DC30 throwback: Offensive Development

Build your own tools for fun and profit



Follow us

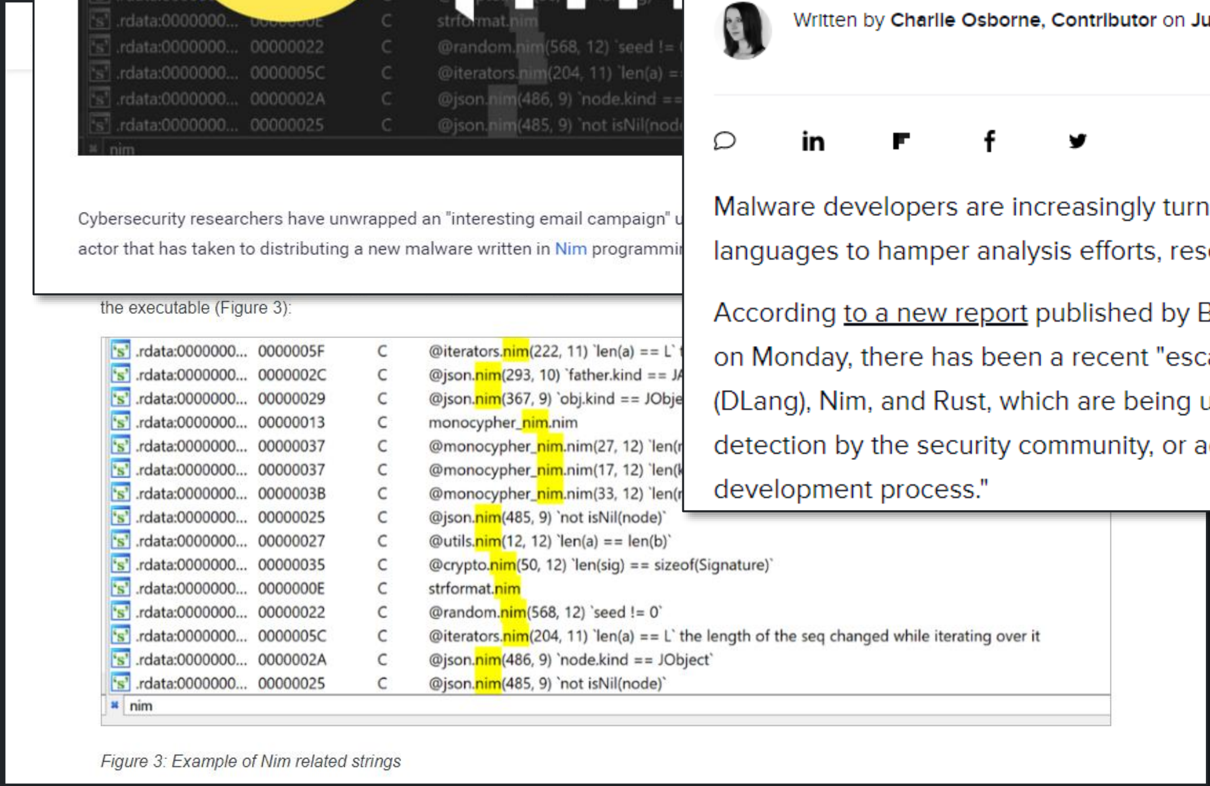
The Hacker News

Home · Cyber Attacks · Vulnerabilities · Offers · Contact

Researchers Spotted Malware Written in Programming Language

March 12, 2021 · Ravie Lakshmanan

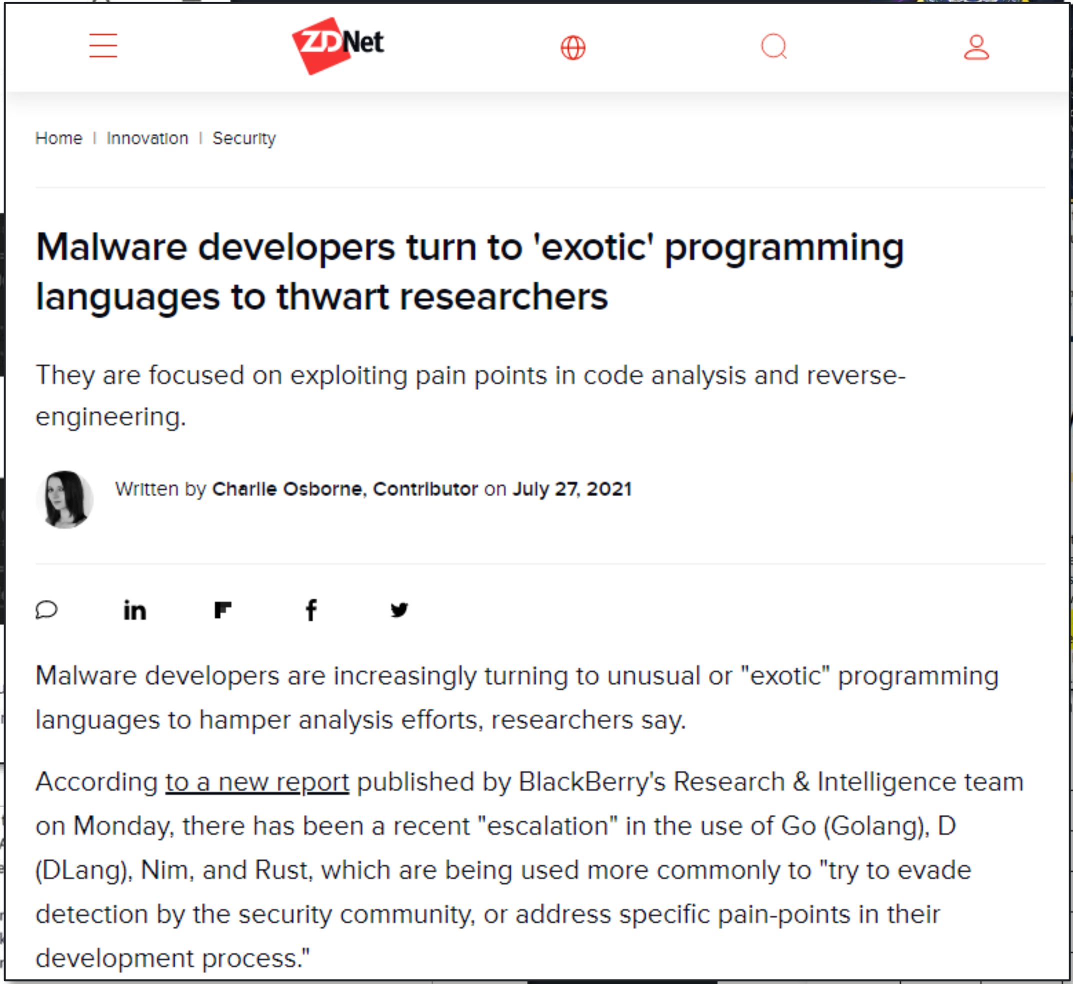
Cybersecurity researchers have unwrapped an "interesting email campaign" u actor that has taken to distributing a new malware written in Nim programmi



the executable (Figure 3):

```
.rdata:0000000...  0000005F   C   @iterators.nim(222, 11) `len(a) == L` t
.rdata:0000000...  0000002C   C   @json.nim(293, 10) `father.kind == JA
.rdata:0000000...  00000029   C   @json.nim(367, 9) `obj.kind == JObje
.rdata:0000000...  00000013   C   monocypher_nim.nim
.rdata:0000000...  00000037   C   @monocypher_nim.nim(27, 12) `len(r
.rdata:0000000...  00000037   C   @monocypher_nim.nim(17, 12) `len(k
.rdata:0000000...  0000003B   C   @monocypher_nim.nim(33, 12) `len(r
.rdata:0000000...  00000025   C   @json.nim(485, 9) `not isNil(node)`
.rdata:0000000...  00000027   C   @utils.nim(12, 12) `len(a) == len(b)`
.rdata:0000000...  00000035   C   @crypto.nim(50, 12) `len(sig) == sizeof(Signature)`
.rdata:0000000...  0000000E   C   strformat.nim
.rdata:0000000...  00000022   C   @random.nim(568, 12) `seed != 0`
.rdata:0000000...  0000005C   C   @iterators.nim(204, 11) `len(a) == L` the length of the seq changed while iterating over it
.rdata:0000000...  0000002A   C   @json.nim(486, 9) `node.kind == JObject`
.rdata:0000000...  00000025   C   @json.nim(485, 9) `not isNil(node)`
nim
```

Figure 3: Example of Nim related strings



ZDNet

Home | Innovation | Security

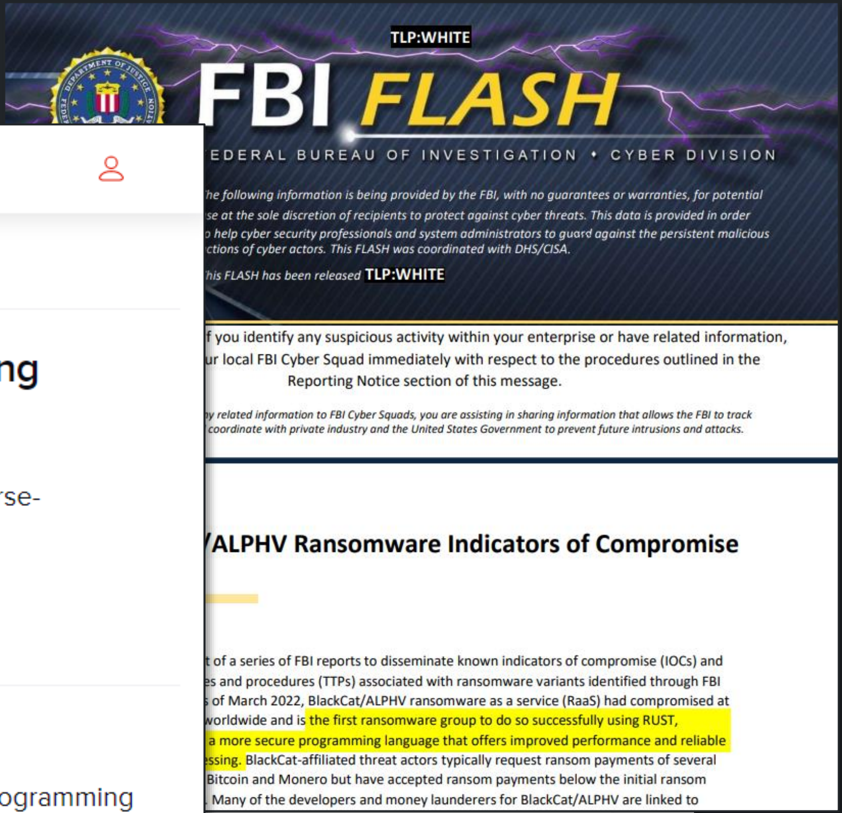
Malware developers turn to 'exotic' programming languages to thwart researchers

They are focused on exploiting pain points in code analysis and reverse-engineering.

Written by **Charlie Osborne, Contributor** on **July 27, 2021**

Malware developers are increasingly turning to unusual or "exotic" programming languages to hamper analysis efforts, researchers say.

According to a new report published by BlackBerry's Research & Intelligence team on Monday, there has been a recent "escalation" in the use of Go (Golang), D (DLang), Nim, and Rust, which are being used more commonly to "try to evade detection by the security community, or address specific pain-points in their development process."



TLP:WHITE

FBI FLASH

EDERAL BUREAU OF INVESTIGATION • CYBER DIVISION

he following information is being provided by the FBI, with no guarantees or warranties, for potential se at the sole discretion of recipients to protect against cyber threats. This data is provided in order o help cyber security professionals and system administrators to guard against the persistent malicious ctions of cyber actors. This FLASH was coordinated with DHS/CISA.

his FLASH has been released **TLP:WHITE**

f you identify any suspicious activity within your enterprise or have related information, ur local FBI Cyber Squad immediately with respect to the procedures outlined in the Reporting Notice section of this message.

y related information to FBI Cyber Squads, you are assisting in sharing information that allows the FBI to track coordinate with private industry and the United States Government to prevent future intrusions and attacks.

/ALPHV Ransomware Indicators of Compromise

t of a series of FBI reports to disseminate known indicators of compromise (IOCs) and es and procedures (TTPs) associated with ransomware variants identified through FBI s of March 2022, BlackCat/ALPHV ransomware as a service (RaaS) had compromised at worldwide and is the first ransomware group to do so successfully using RUST, a more secure programming language that offers improved performance and reliable ssing. BlackCat-affiliated threat actors typically request ransom payments of several Bitcoin and Monero but have accepted ransom payments below the initial ransom . Many of the developers and money launderers for BlackCat/ALPHV are linked to

g Malware Count Over Time

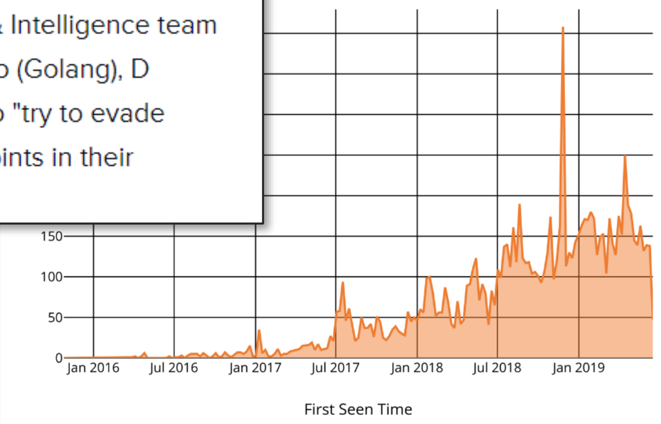


Figure 1. Timeline of Go Malware samples based on first seen dates.

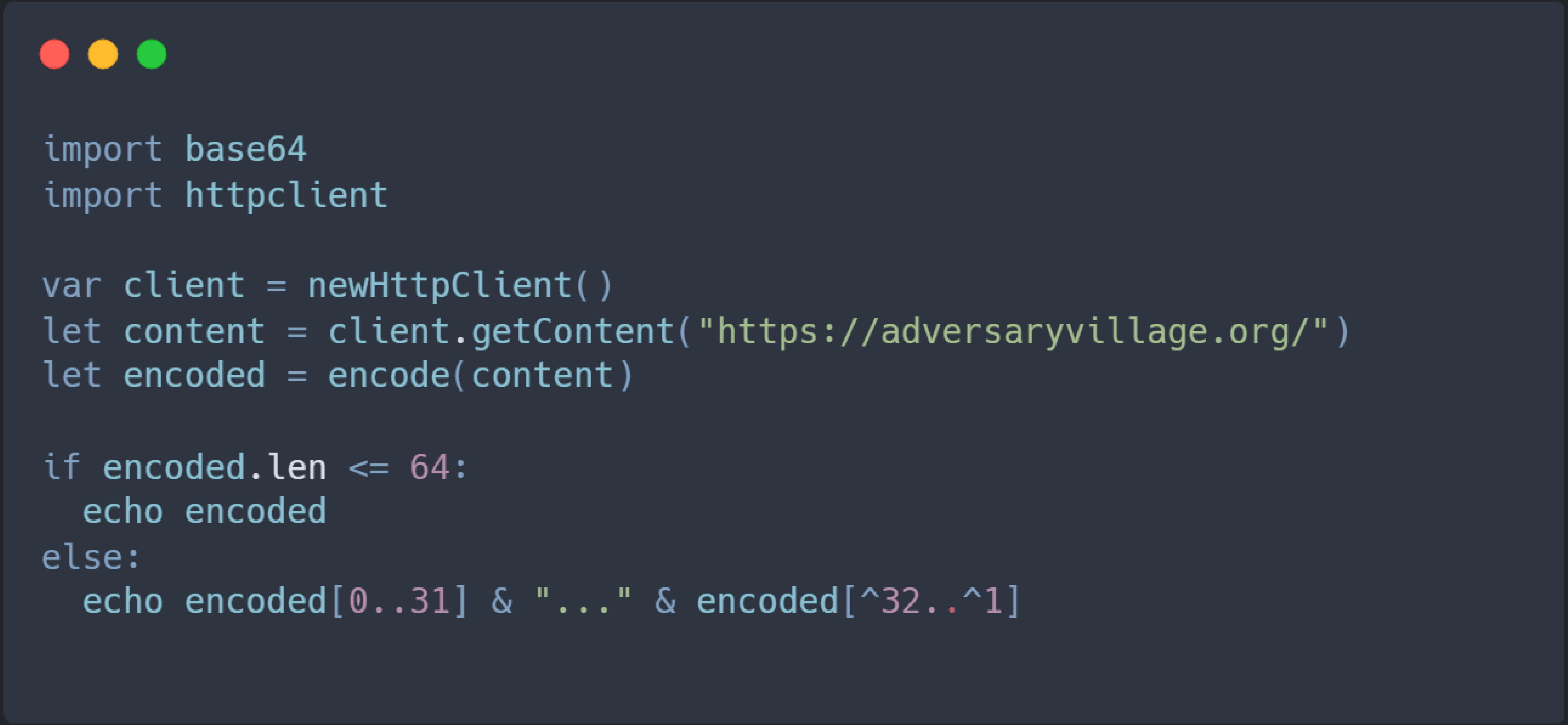
02 | Nim

Nim for malware development

- Compiles directly to C, C++, Objective-C or Javascript
- Doesn’t rely on VM or runtime, yields small binaries
- Python-inspired syntax, rapid development and prototyping
  - Avoids you having to write C/C++ (goodbye vulns!)
- Has an extremely mature Foreign Function Interface (FFI)
- Super easy cross-compilation (using mingw)

03 | Nim in Practice

Getting acquainted with the syntax



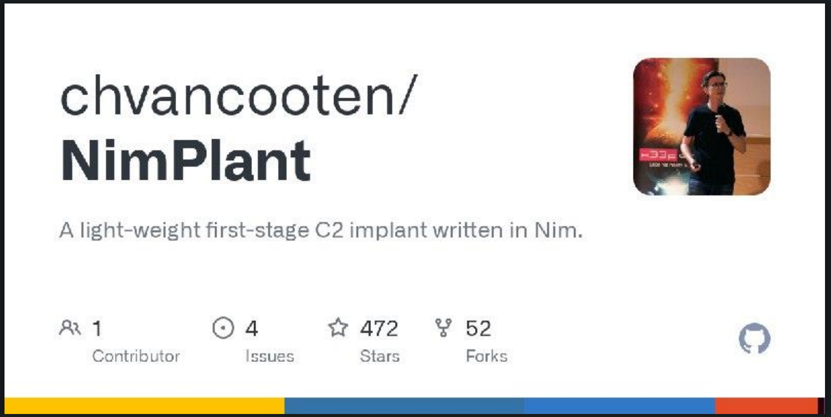
```nim
import base64
import httpclient

var client = newHttpClient()
let content = client.getContent("https://adversaryvillage.org/")
let encoded = encode(content)

if encoded.len <= 64:
  echo encoded
else:
  echo encoded[0..31] & "..." & encoded[^32..^1]
```

rosettacode.org/wiki/Base64_encode_data#Nim

04 | Getting Hands-On

Nimplant: A lightweight stage-one C2

- C2 implant in Nim, server in Python
- Web GUI in Next.JS
- Designed for early-access operations
- Configurable HTTP C2 behavior
- Less suspicious due to native implementations
- Support for BOFs, inline execute-assembly, dynamic shellcode invocation, and more



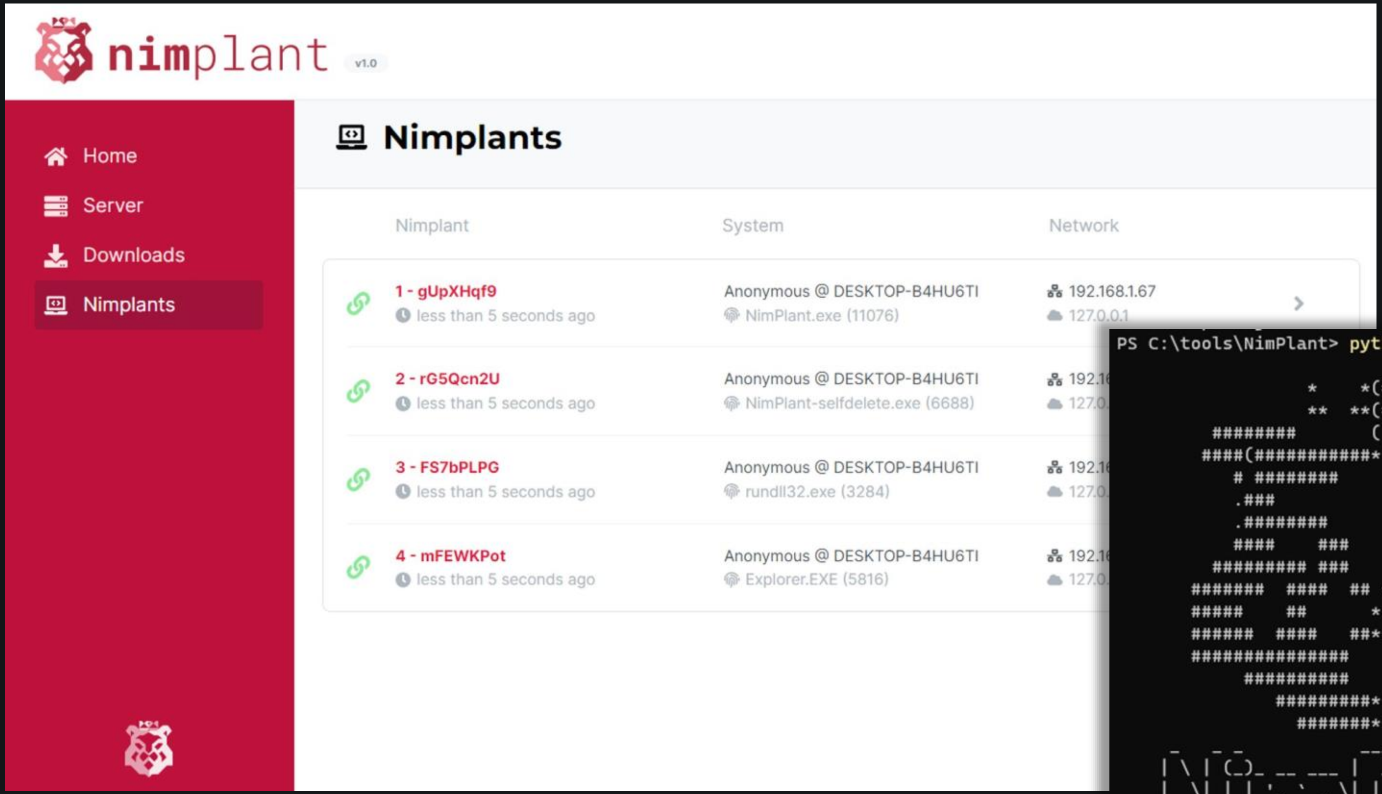
chvancooten/
NimPlant

A light-weight first-stage C2 implant written in Nim.

1	4	472	52
Contributor	Issues	Stars	Forks

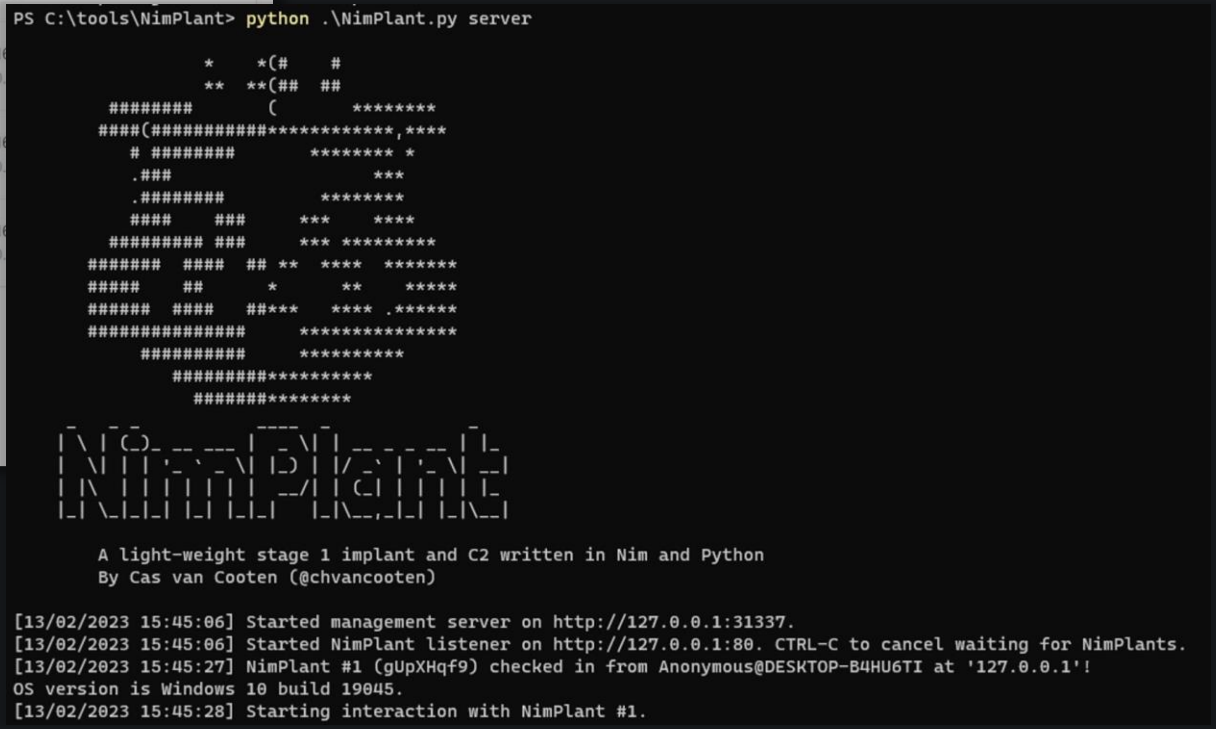
04 | Getting Hands-On

Nimplant: A lightweight stage-one C2



Nimplants

Nimplant	System	Network
1 - gUpXHqf9 less than 5 seconds ago	Anonymous @ DESKTOP-B4HU6TI NimPlant.exe (11076)	192.168.1.67 127.0.0.1
2 - rG5Qcn2U less than 5 seconds ago	Anonymous @ DESKTOP-B4HU6TI NimPlant-selfdelete.exe (6688)	192.16 127.0.
3 - FS7bPLPG less than 5 seconds ago	Anonymous @ DESKTOP-B4HU6TI rundll32.exe (3284)	192.16 127.0.
4 - mFEWKPot less than 5 seconds ago	Anonymous @ DESKTOP-B4HU6TI Explorer.EXE (5816)	192.16 127.0.



```
PS C:\tools\NimPlant> python .\NimPlant.py server

                  *    *(#    #
                  **  **(##  ##
         ########       (        ********
        ####(###########************,****
           # ########       ******** *
           .###                   ***
           .########        ********
           ####    ###     ***    ****
         ######### ###     *** *********
       #######  ####  ## **  ****  *******
       #####    ##      *      **    *****
       ######  ####   ##***   **** .******
       ###############     ***************
            ##########     **********
               #########**********
                 #######********

     _   _ _            ____  _             _
    | \ | (_)_ __ ___  |  _ \| | __ _ _ __ | |_
    |  \| | | '_ ` _ \ | |_) | |/ _` | '_ \| __|
    | |\  | | | | | | ||  __/| | (_| | | | | |_
    |_| \_|_|_| |_| |_||_|   |_|\__,_|_| |_|\__|

        A light-weight stage 1 implant and C2 written in Nim and Python
        By Cas van Cooten (@chvancooten)

[13/02/2023 15:45:06] Started management server on http://127.0.0.1:31337.
[13/02/2023 15:45:06] Started NimPlant listener on http://127.0.0.1:80. CTRL-C to cancel waiting for NimPlants.
[13/02/2023 15:45:27] NimPlant #1 (gUpXHqf9) checked in from Anonymous@DESKTOP-B4HU6TI at '127.0.0.1'!
OS version is Windows 10 build 19045.
[13/02/2023 15:45:28] Starting interaction with NimPlant #1.
```