





\$whoami

- **What I do ?** Threat [**Detection** | **Hunting** | **Intelligence**]

- **Where ?** Seqrite Labs , Quick Heal , India



[@ElementalX2](#)



[Subhajeet Singha](#)





Agenda

01 Contents.

02 Rise of Trend.

03 Fall of Trend.

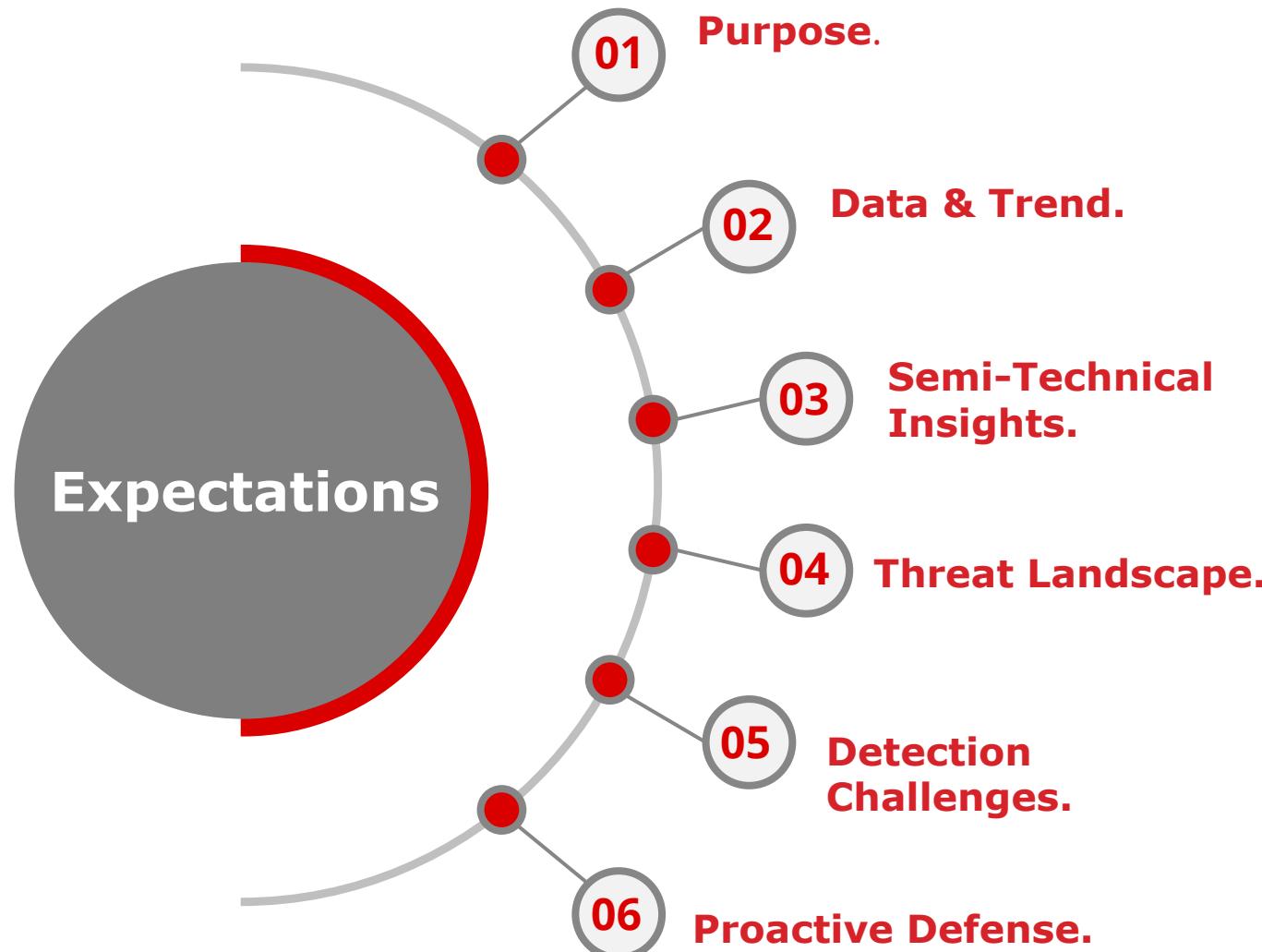
04 Pseudo-Rise.

05 Tool Demo : Go-Peep.



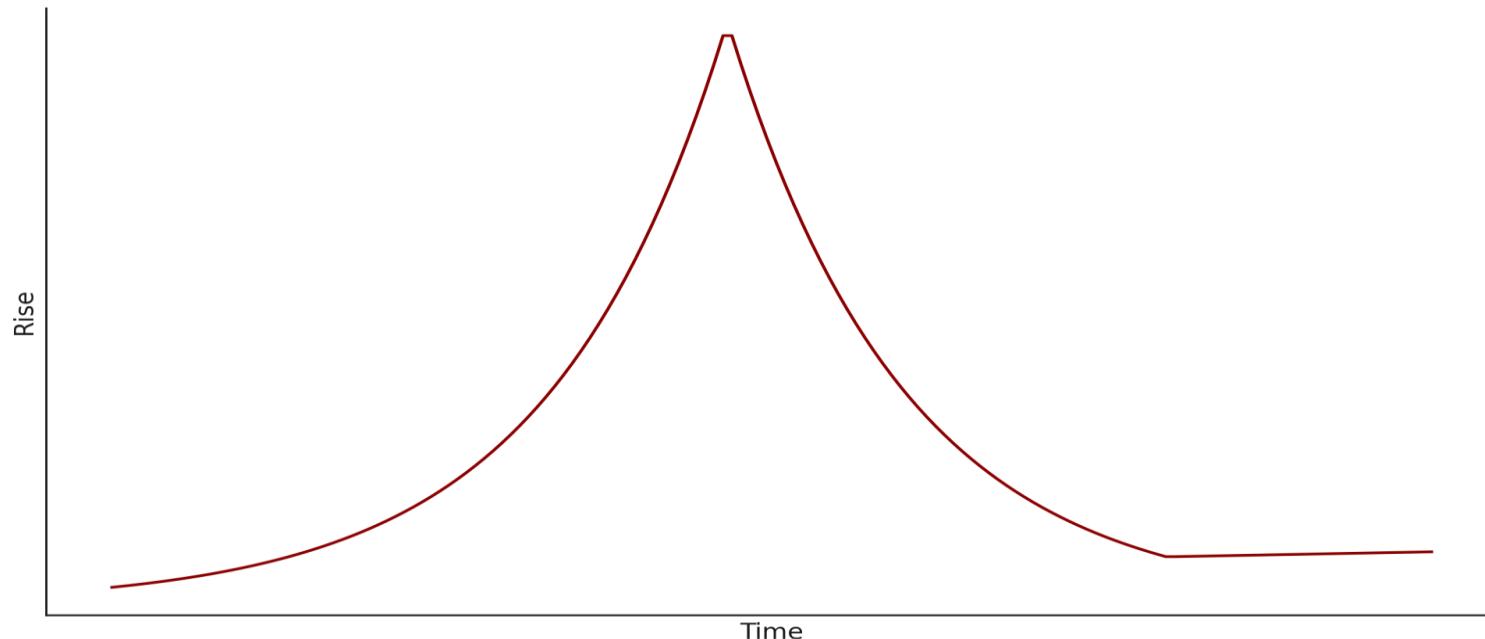


CONTENTS



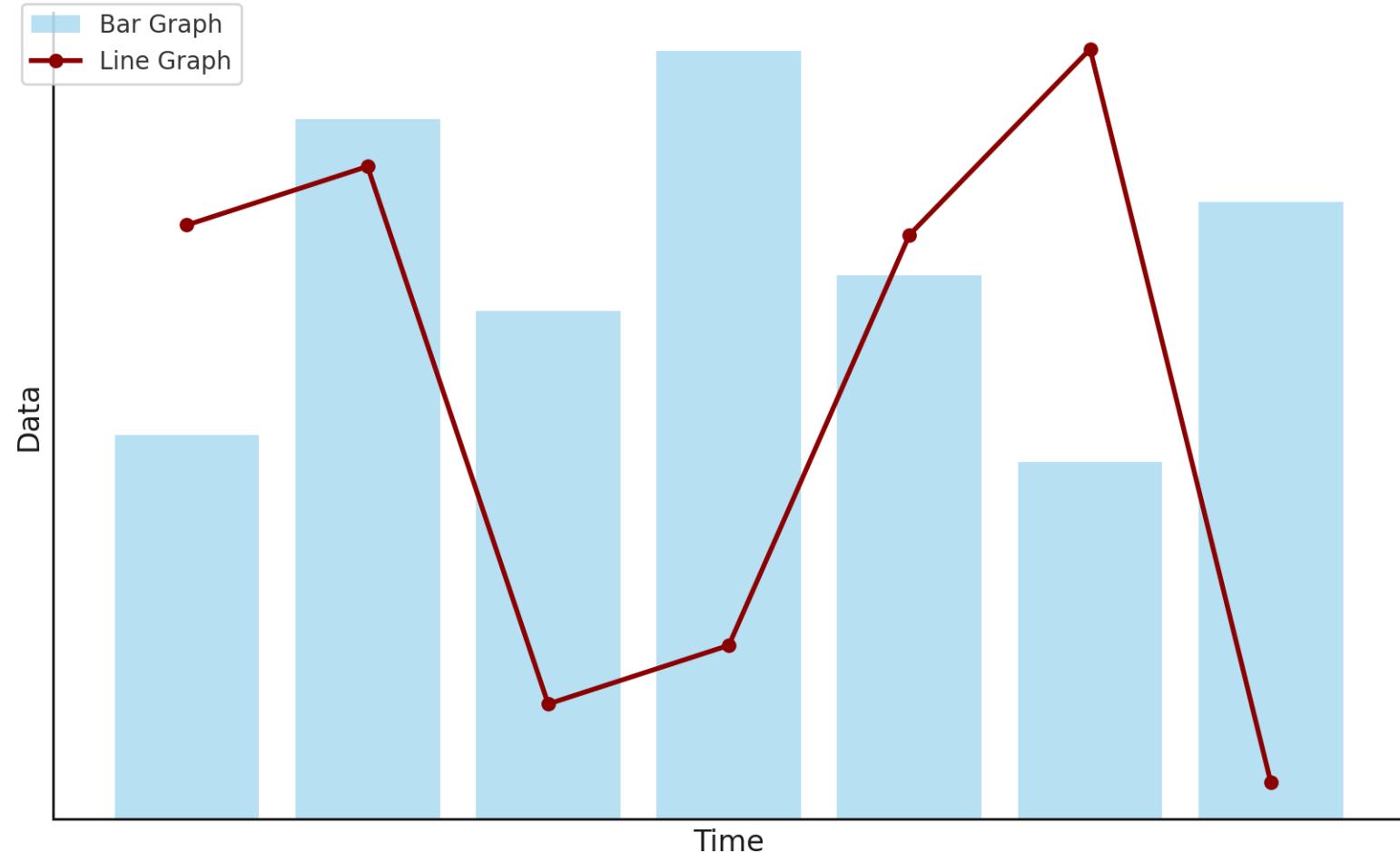
01

Purpose : Highlight the goal of understanding Golang-based malware trends and its implications for malware defense.



02 Data & Trend

Provide statistical data showing the rise of Golang malware and geographic distribution trends.



03 Semi-technical Insights

Offer insights into the semi-technical aspects of Golang malware, including code characteristics, libraries, and structures that have influenced its rise.



04 Threat Landscape

Explore the impact of Golang malware on global cybersecurity, specifically addressing notable campaigns (state-sponsored and ransomware) and widespread infections



05 **Detection Challenges**

Explore how the anti-malware community and other vendors have faced a lot of challenges and how they did overcome from this.



06 Proactive Defense

Detail advancements in early detection techniques, specifically focusing on open-source repositories and tools like Go-Obfuscator, Garble that are aiding Golang malware developers giving a pseudo-rise.

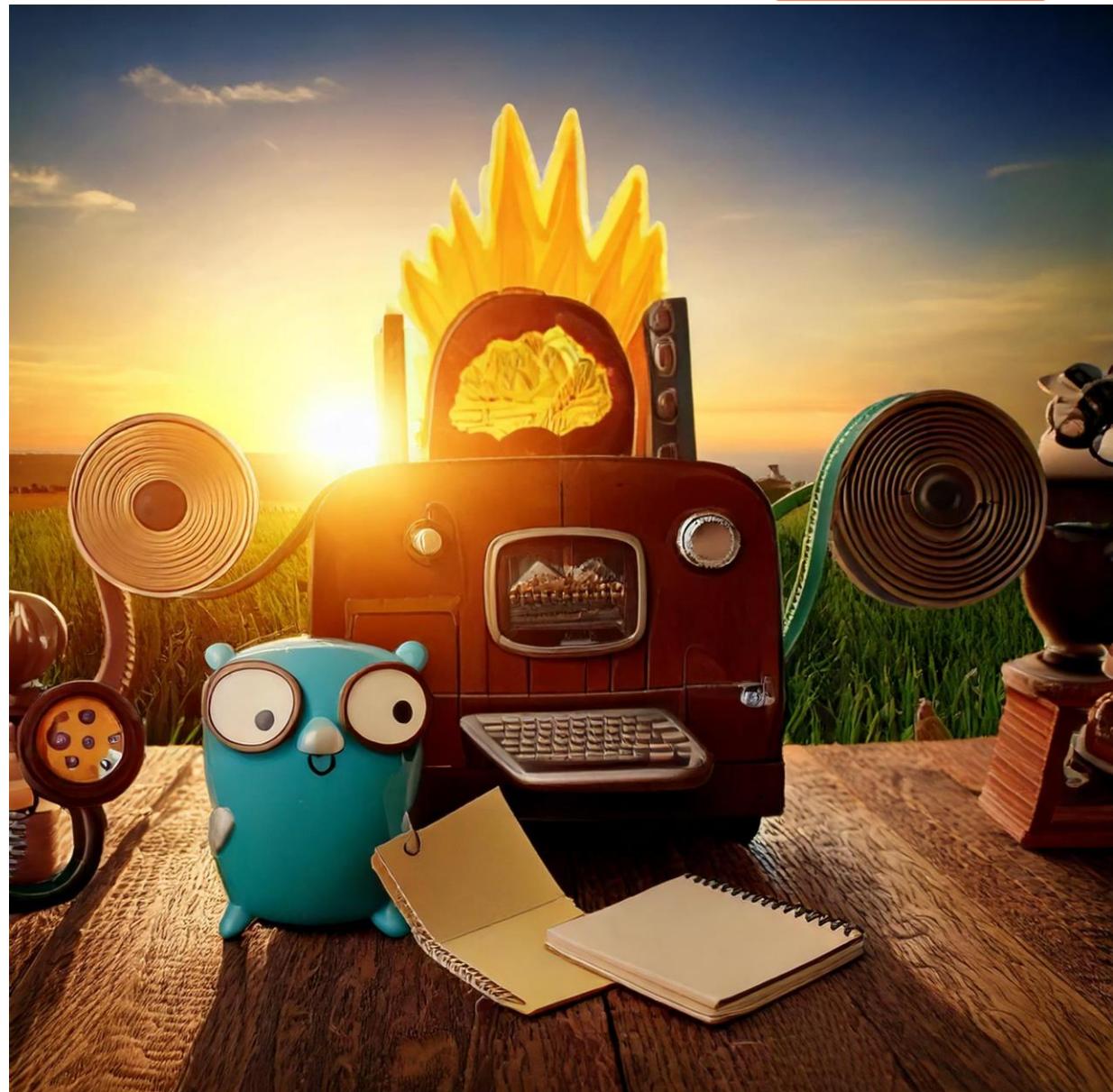




CASE STUDY : RISE OF TREND

CLAIMS FOR "RISE OF GOLANG MALWARE"

- Increased Prevalence of Golang Malware.
- Diverse Use Cases of Golang in Cyber Threats.
- Geographical Spread of Golang Malware[APT].
- Impact of Golang in High-Profile Threat Campaigns.
- Accessibility and Flexibility Fueling Malware Development.



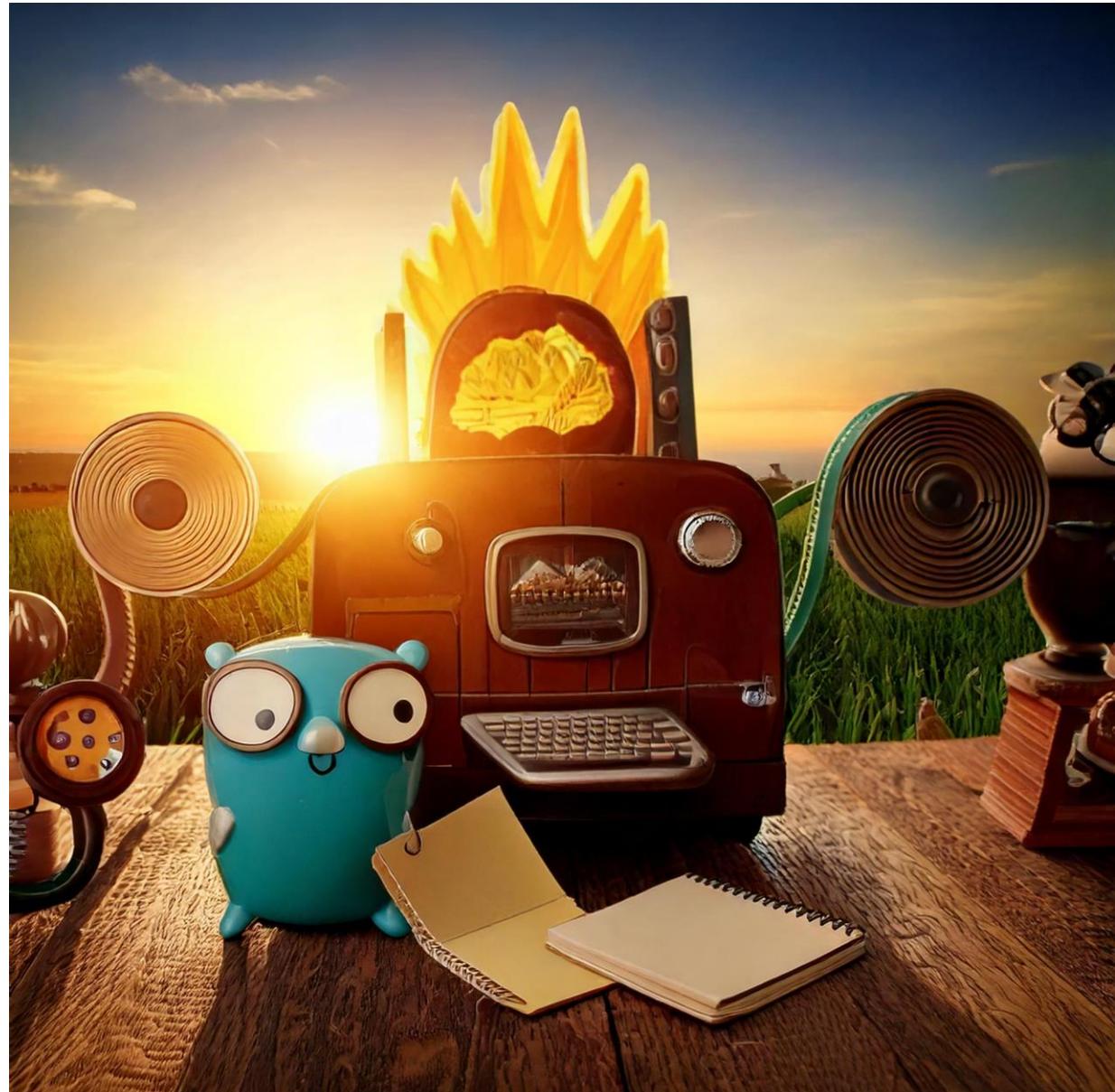


CASE STUDY : RISE OF TREND

SEQRITE

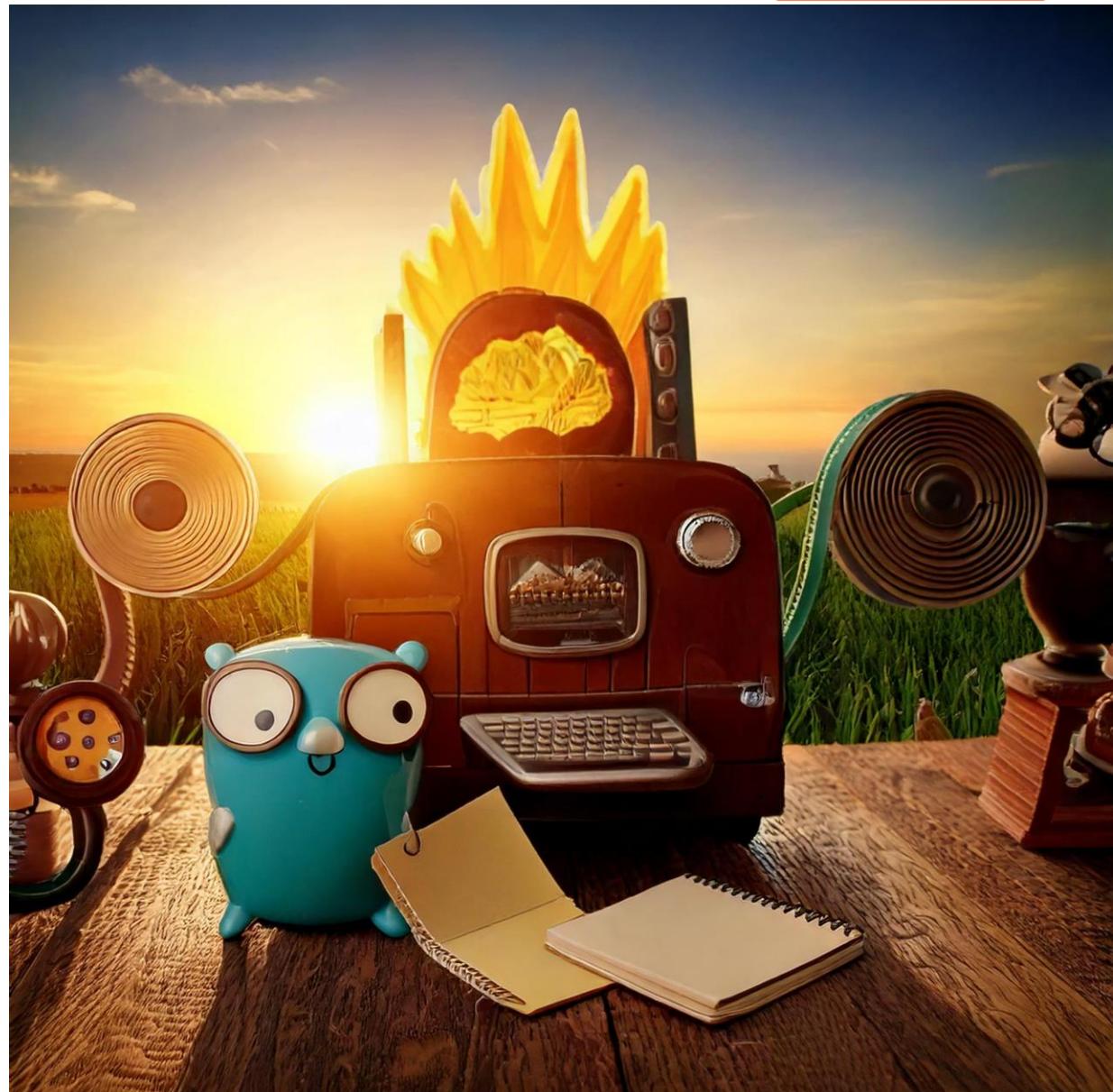
Quick Heal

Time to debunk ?



Claim: Increased Prevalence of Golang Malware.

"In the past five years, there has been a significant rise in Golang-based malware In-The-Wild (ITW), indicating a growing trend in its adoption by threat actors."





CASE STUDY : RISE OF TREND

SEQRITE

Quick Heal

Malware Uses Google Go Language



Migration User

Sep 18, 2012 05:50 PM

Designed in 2007 and introduced in late 2009, the Go programming language developed by Google has been gaining momentum the past three years. It is now being used to develop malware. Recently seen in the wild, [Trojan.Encryoko](#) is a new threat associated with components which are written in Go. The Trojan attempts to encrypt various file formats on compromised computers, rendering the encrypted files unusable.

The original sample we acquired, a file named GalaxyNxRoot.exe, is actually a dropper written in .NET which disguises itself as a [rooting tool](#) to trick users into installing it.

Where it all started?



Andrew LeFevre
@capnspacelhook

Prediction for 2019: Golang malware/tradecraft will get much bigger. It easily cross compiles, is easy to write in, and the binaries it produces are just so different than typical C/C++ binaries and it seems AV/EDR doesn't know what to do with it yet.

9:57 PM · Dec 31, 2018

200 Reposts 23 Quotes 678 Likes 41 Bookmarks

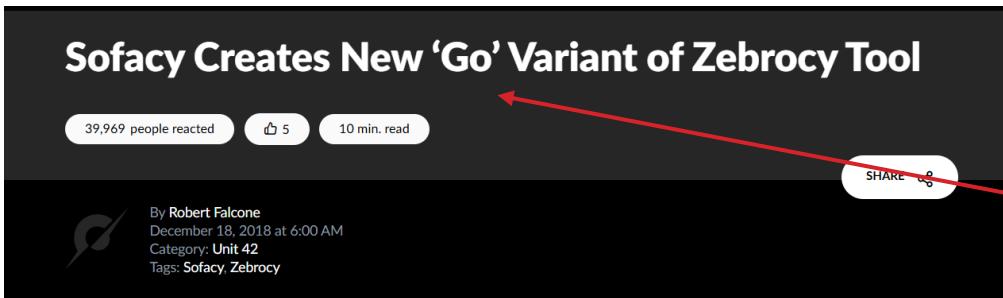
Andrew said it before it was cool!

Sofacy Creates New 'Go' Variant of Zebrocy Tool

39,969 people reacted | 5 | 10 min. read

By Robert Falcone
December 18, 2018 at 6:00 AM
Category: Unit 42
Tags: Sofacy, Zebrocy

SHARE 9



// AUTHORS

IVAN KWIATKOWSKI | EXPERT FÉLIX AIME | EXPERT PIERRE DELCHER

On December 4, 2019, we discovered watering hole websites that were compromised to selectively trigger a drive-by download attack with fake Adobe Flash update warnings. This campaign has been active since at least **May 2019**, and targets an Asian religious and ethnic group.

The threat actor's unsophisticated but creative toolset has been evolving a lot since the inception date, may still be in development, and leverages Sojson obfuscation, NSIS installer, Python, open-source code, GitHub distribution, **Go language**, as well as Google Drive-based C2 channels.

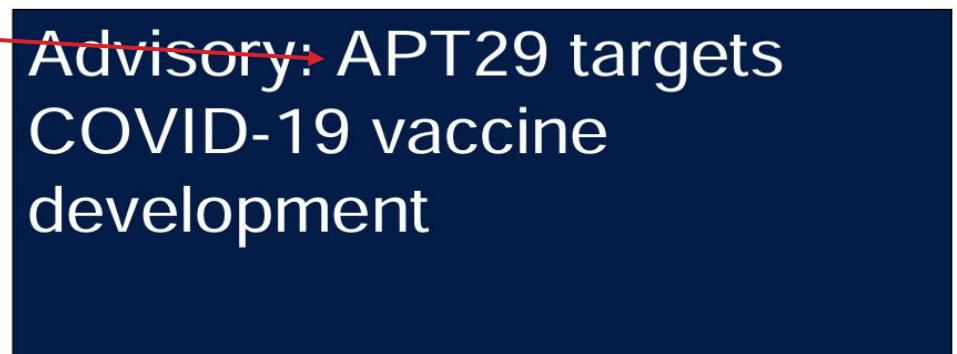
The threat actor's operational target is not clear because, unfortunately, we haven't been able to observe many live operations, and we couldn't identify any overlap with known intrusion sets.

2018 – 2019

Early Notable Trends



Advisory: APT29 targets COVID-19 vaccine development



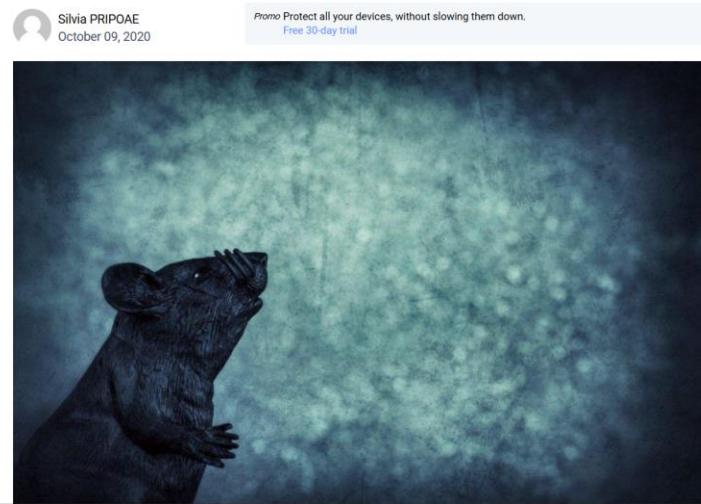
NEWS | THREATS

Analyzing a new stealer written in Golang

Posted: January 30, 2019 by hasherezade

Golang (Go) is a relatively new programming language, and it is not common to find malware written in it. However, new variants written in Go are slowly emerging, presenting a challenge to malware analysts. Applications written in this language are bulky and look much different under

There's a New a Golang-written RAT in Town



Spotted between August to September 2020, SUNSHUTTLE is a Golang-based malware that acts as a command-and-control backdoor, establishing a secure connection with an attacker-controlled server to receive commands to download and execute files, upload files from the system to the server, and execute operating system commands on the compromised machine.

CRIMEWARE

New Snake Ransomware Adds Itself to the Increasing Collection of Golang Crimeware

JIM WALTER / JANUARY 23, 2020

We are just about 1 month into 2020, and so far, there has been no break in the ongoing flurry of new or varied ransomware campaigns. Amongst the well-established families (Ryuk, Maze, REvil) we now have another to add to the list..."Snake".

SentinelLabs has observed the Snake ransomware in targeted campaigns over the last month. While it contains all the hallmarks of standard ransomware,

2020

NEW GOLANG-BASED CRYPTO WORM INFECTS WINDOWS AND LINUX SERVERS

Pierluigi Paganini December 31, 2020

```
Package hello/src/exp: /media/psf/AllFiles/Users/mac/go/src/hello/src/exp
File: <autogenerated>
    init Lines: 1 to 40 (39)
File: exp.go
    NewAttack Lines: 22 to 115 (93)
    registerExp Lines: 115 to 119 (4)
    (*Attack)All Lines: 119 to 129 (10)
    (*Attack)Loop Lines: 129 to 131 (2)
    (*Attack).Loopfunc1 Lines: 137 to 138 (1)
File: jenkins.go
    Jenkins Lines: 14 to 53 (39)
    checkJenkins Lines: 53 to 61 (8)
    init0 Lines: 61 to 66 (5)
File: mysql.go
    Mysql Lines: 15 to 90 (75)
    mysql_max_allowed_packet Lines: 90 to 114 (24)
    sqlExec Lines: 114 to 130 (16)
    init1 Lines: 130 to 139 (9)
```



Abcbot – A New Evolving Wormable Botnet Malware Targeting Linux

Nov 12, 2021 · Ravie Lakshmanan

ANTI-MALWARE RESEARCH · 1 min read ·

Debugging MosaicLoader, One Step at a Time

 Janos Gergo SZELES
July 20, 2021

Promo Protect all your devices, without slowing them down.
Free 30-day trial

LevelBlue Labs finds new Golang malware (BotenaGo) targeting millions of routers and IoT devices with more than 30 exploits

November 11, 2021 | Ofer Caspi

DECAF Ransomware: A New Golang Threat Makes Its Appearance

Posted by Hido Cohen & Michael Dereviashkin on October 28, 2021

2021

TA505 adds GoLang crypter for delivering miners and ServHelper

 Jason Reaves · Follow
Published in Walmart Global Tech Blog · 6 min read · Jul 6, 2021



CASE STUDY : RISE OF TREND

SEQRITE

Quick Heal

TellYouThePass Ransomware Analysis Reveals a Modern Reinterpretation Using Golang

January 11, 2022 | Anmol Maurya | Endpoint Security & XDR



HOME > CYBERSECURITY NEWS

Agenda, a New Golang Ransomware, Is on the Loose

The Ransomware Can Be Customized for Every Victim.

Last updated on October 20, 2022



2022

Arid Gopher - A New Golang Malware Spotted in the Town

Malware and Vulnerabilities • March 24, 2022 • Cyware Alerts - Hacker News



CRIMEWARE

BLOG

SECURONIX THREAT LABS SECURITY ADVISORY: NEW GOLANG ATTACK CAMPAIGN GO#WEBBFUSCATOR LEVERAGES OFFICE MACROS AND JAMES WEBB IMAGES TO INFECT SYSTEMS

THREAT RESEARCH

CrateDepression | Rust Supply-Chain Attack Infects Cloud CI Pipelines with Go Malware

👤 JUAN ANDRÉS GUERRERO-SAADE / 📅 MAY 19, 2022

By Juan Andrés Guerrero-Saade & Phil Stokes



CASE STUDY : RISE OF TREND

SEQRITE

Quick Heal

ADVERSARY

DragonSpark | Attacks Evade Detection with SparkRAT and Golang Source Code Interpretation

By Aleksandar Milenkoski / JANUARY 24, 2023

By Aleksandar Milenkoski, Joey Chen, and Amitai Ben Shushan Ehrlich

This new malware targets Discord and browser data using one major unique feature

News By Craig Hale published 15 June 2023

Use of Go programming language is especially worrying, experts warn

Skuld: The Info stealer that Speaks Golang

By Ernesto Fernández Provecho · June 13, 2023

In May 2023, the Trellix Advanced Research Center discovered a new Golang stealer, known as Skuld, that compromised systems worldwide, something that security researchers had also noticed.

The usage of Golang, also known as Go, in malware development is still rare compared to other programming languages. But it has gained significant popularity in recent years due to simplicity, efficiency, and cross-platform compatibility, which lets malware creators target a wide range of operating systems, broadening their potential victim pool. Additionally, Golang's compiled nature lets malware authors produce binary executables that are more challenging to analyze and reverse engineer. This makes it harder for security researchers and traditional anti-malware solutions to detect and mitigate these threats effectively.

This new malware strain tries to steal sensitive information from its victims. To accomplish this task, it searches for data stored in applications such as Discord and web browsers; information from the system and files stored in the victim's folders. Some samples even include a module to steal cryptocurrency assets, which we believe is still in development.

Detailed Analysis of AlphaSeed, a new version of Kimsuky's AppleSeed written in Golang

S2W · Follow
Published in S2W BLOG · 19 min read · May 17, 2023

Author: BLKSMTH | S2W TALON

Last Modified: May 17, 2023



2023

Titan Stealer: A New Golang-Based Information Stealer Malware Emerges

Jan 30, 2023 · Ravie Lakshmanan

Threat Detection / Malware



CASE STUDY : RISE OF TREND

SEQRITE

Quick Heal

BLOG

Operation FlightNight: Indian Government Entities and Energy Sector Targeted by Cyber Espionage Campaign

Arda Büyükkaya · March 27, 2024

APT group GoldenJackal deploys backdoors to air-gapped systems

News

09 Oct 2024 · 7 mins

Kimsuky disguised as a Korean company signed with a valid certificate to distribute Troll Stealer (English ver.)



S2W · Follow

Published in S2W BLOG · 14-min read · Feb 7, 2024

P4nd3m1cb0 · 1/4 It looks like Brazilian #threat actors are trying to work in something new with #golang 😲

Go build ID: "9e71K3Bnkt22jTtd3-BN/oPOrbzV96aoE40e45p7/oZ7E1nLQ4KsUAa4NWAz5/XqDJz5ptN7RdLxuuZi1y"

#cti #ThreatHunting #cybersecurity #IOC #trojan #banker #malware

5:27 AM · Jul 5, 2024 · 3,068 Views

2024

Analysis of APT-C-00 (OceanLotus) Double Loader and Related VMP Loader

NetmanageIT OpenCTI - opencsti.netmanageit.com



Daniel Bender

Sep 24, 2024 · 1 min read



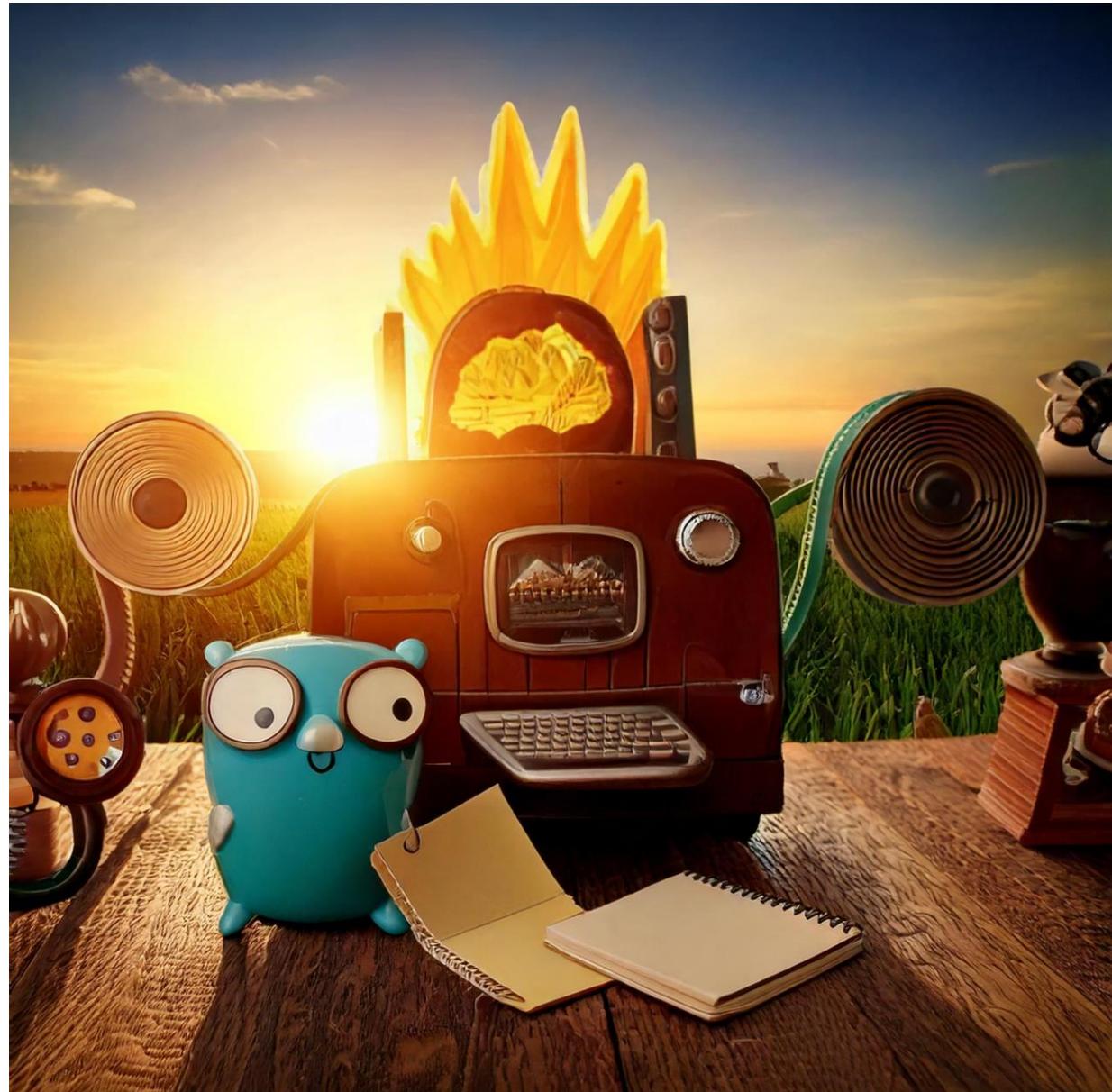
CASE STUDY : RISE OF TREND

SEQRITE

Quick Heal

Claim: Diverse Use Cases of Golang in Cyber Threats.

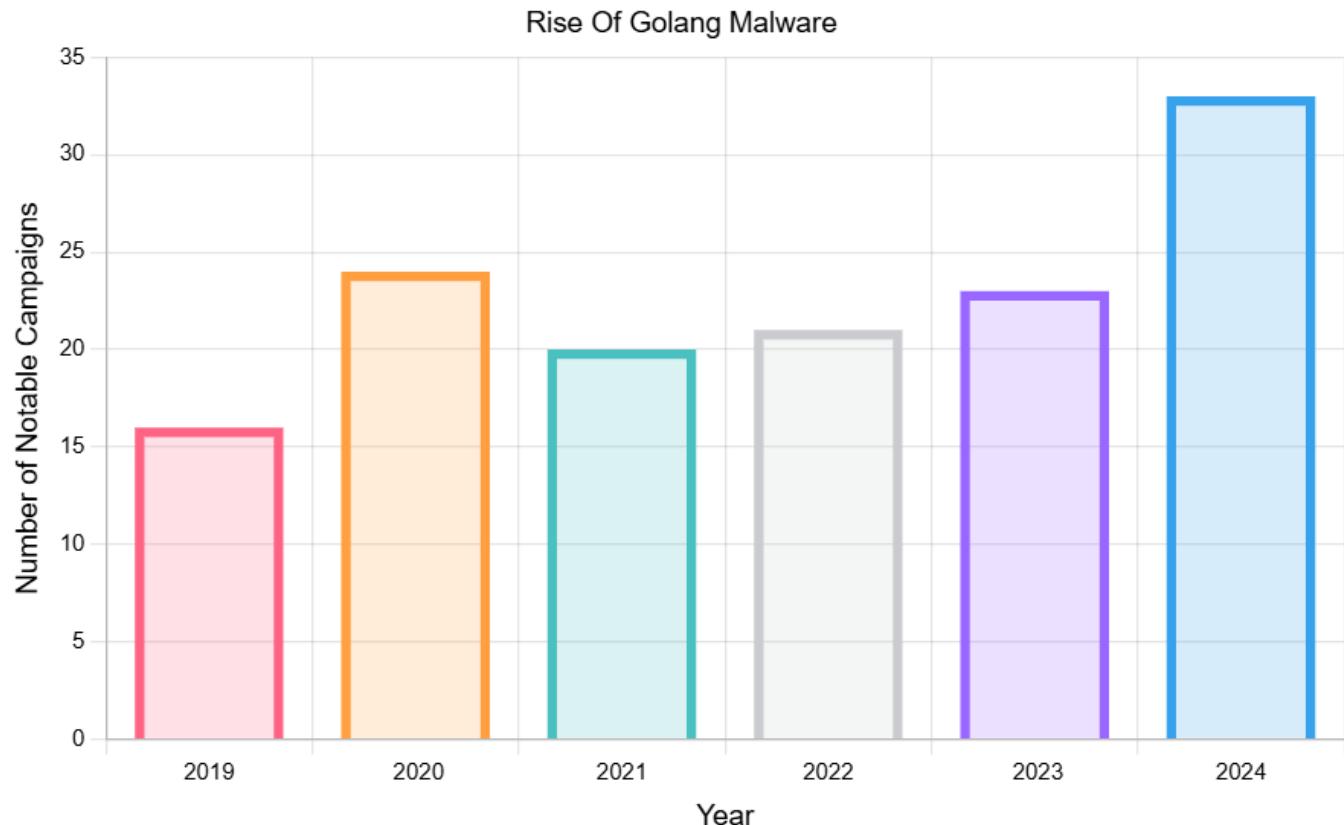
"In the past five years, Golang has powered malware across ransomware, APTs, stealers, and RATs."





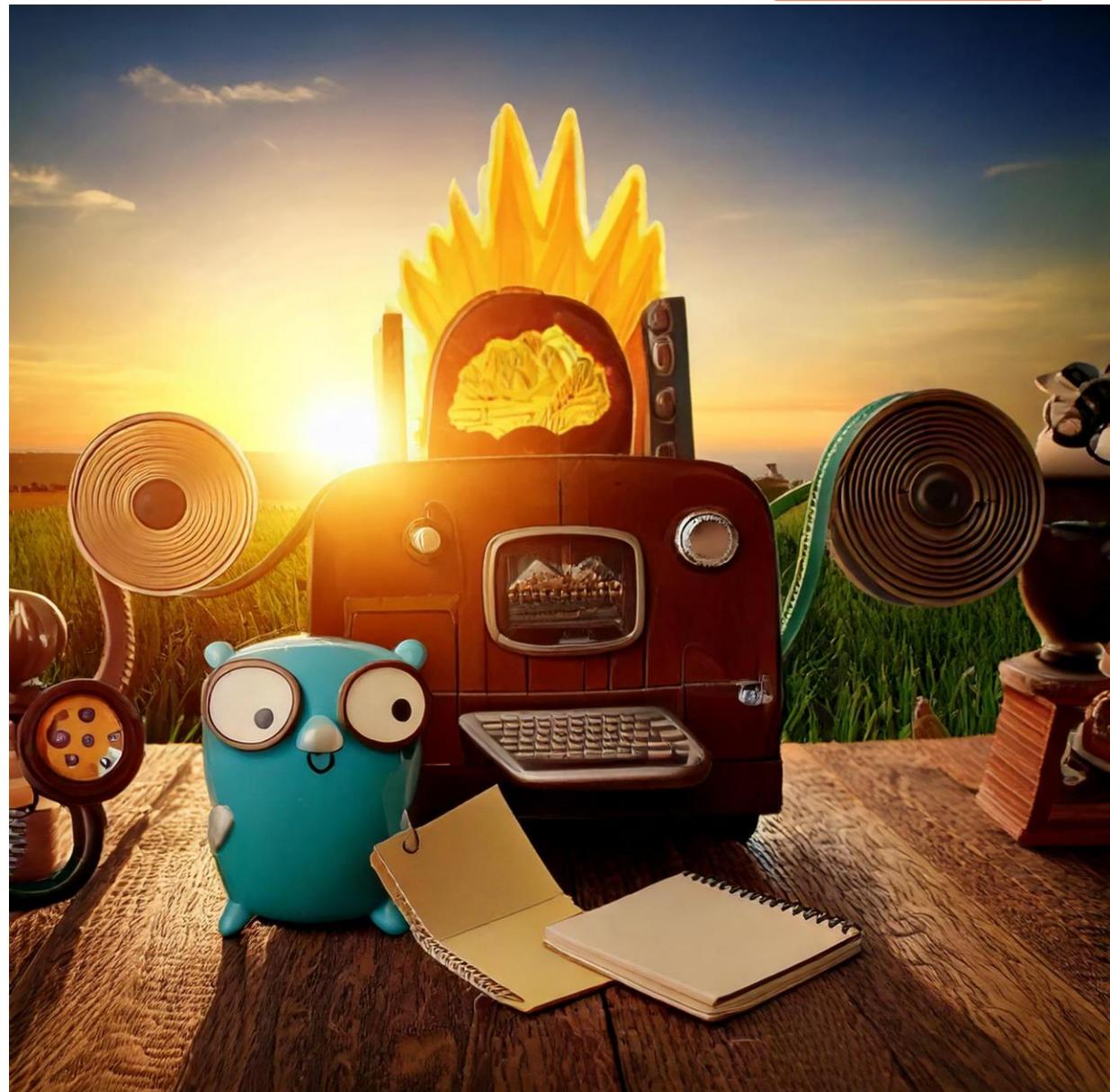
CASE STUDY : RISE OF TREND

- This graph has been plotted solely on the based on some interesting and notable reports of various vendors and from sources like AlienVault OTX & Malpedia only which includes campaigns related to :
 1. Advanced Persistent Threat Groups.
 2. Stealers / RATs.
 3. Botnets.
 4. Ransomware
 5. Miners.
 6. Wipers.



Claim: Geographical Spread of Golang Malware[APT].

"In the past five years, Golang has been welcomed by Advanced Persistent Threat Groups across the globe"





CASE STUDY : RISE OF TREND



- [APT28](#).
- [APT29](#).



- [TA416](#)
- Evasive Panda
- [Earth Lusca](#)



- [Kimsuky](#). [\[1\]](#) [\[2\]](#)
- [Lazarus](#).



CASE STUDY : RISE OF TREND



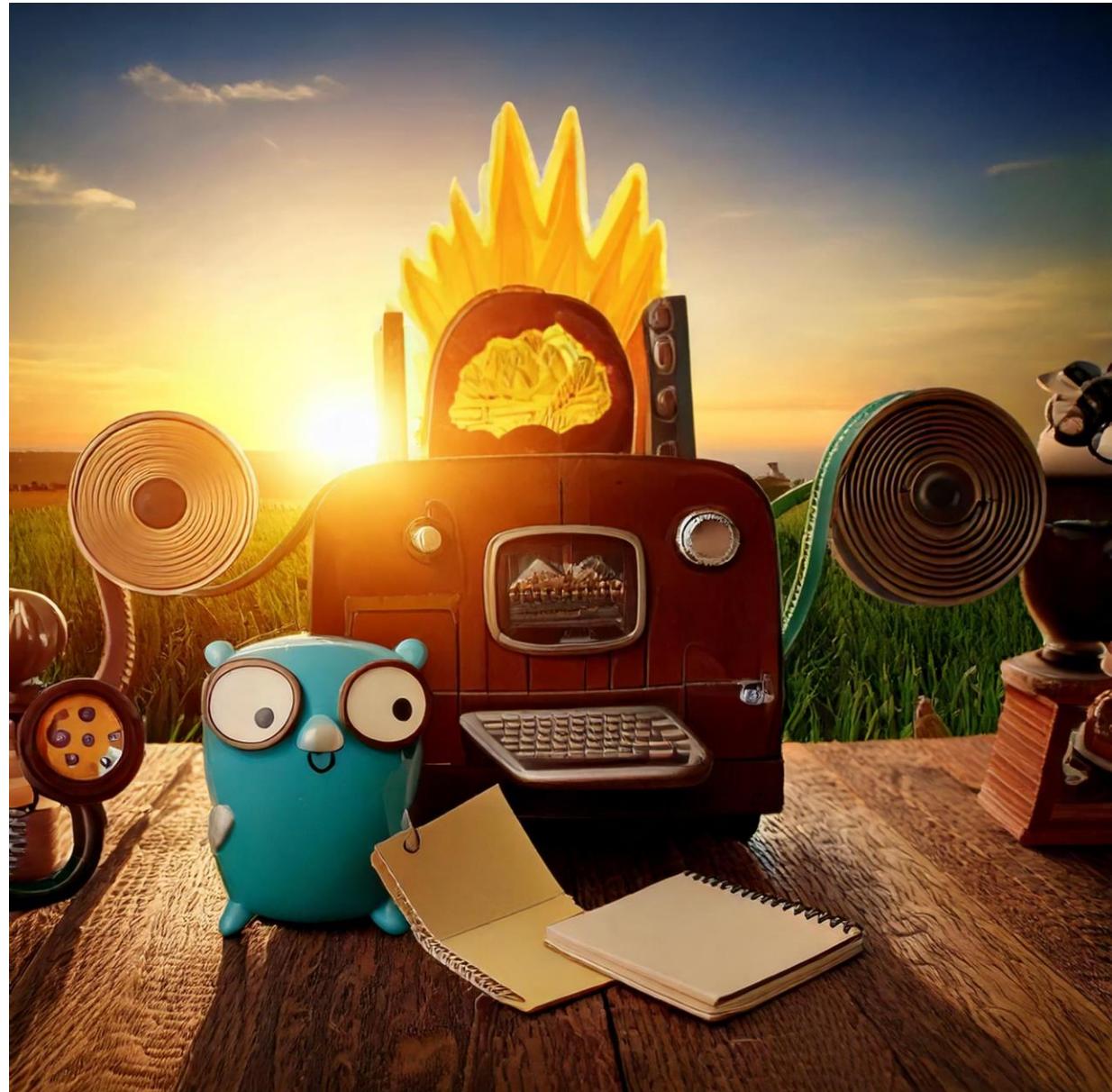
- [MuddyWater.](#)
- [APT34.](#)



- [Arid Viper](#)

Claim: Impact of Golang in High-Profile Threat Campaigns.

"In the past five years, there has been some ongoing tension across the globe, we will see how Golang has been leveraged by threat groups or actors to further execute their campaigns."





CASE STUDY : RISE OF TREND

SolarWinds Incident



GoldMax (aka SUNSHUTTLE), which was discovered by Microsoft and FireEye (now Mandiant) in March 2021, is a [Golang-based malware](#) that acts as a command-and-control backdoor, establishing a secure connection with a remote server to execute arbitrary commands on the compromised machine.

RU-UA Geopolitical Tensions.



HermeticWiper | New Destructive Malware Used In Cyber Attacks on Ukraine

by JUAN ANDRÉS GUERRERO-SAADE / FEBRUARY 23, 2022

This post was updated Feb 28th 2022 to include new IOCs and the PartyTicket 'decoy ransomware'.

IND-PK Geopolitical Tensions.



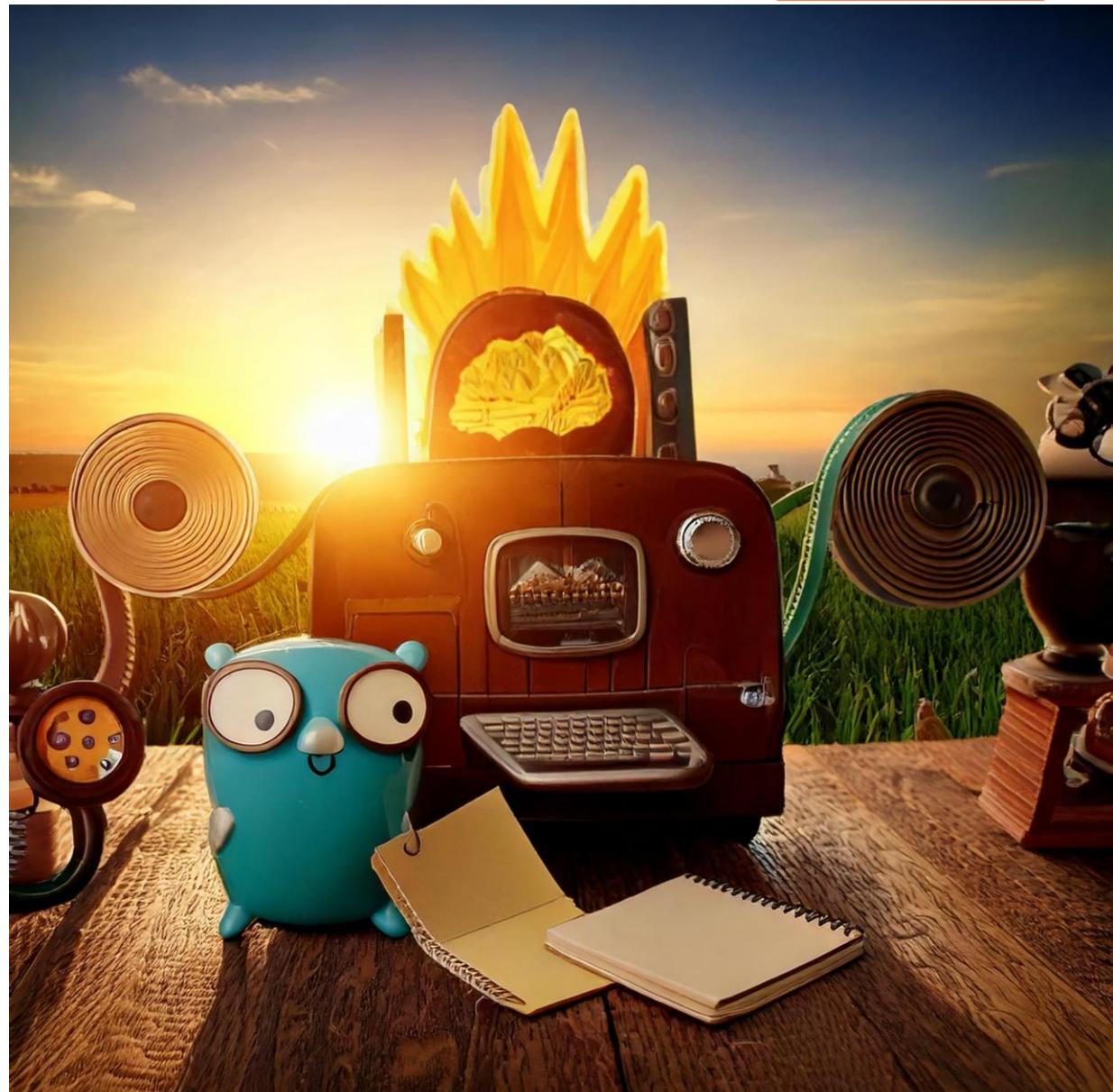
Transparent Tribe Targets Indian Government, Defense, and Aerospace Sectors Leveraging Cross-Platform Programming Languages

RESEARCH & INTELLIGENCE / 05.22.24 / The BlackBerry Research and Intelligence Team



Claim: Accessibility and Flexibility Fueling Malware Development.

"In the past five years, Golang malware development has been accessible in means of Open-Source Software development projects and Command and Control software, with its flexibility fueling widespread adoption."





CASE STUDY : RISE OF TREND

- Software Development Projects ([gopsutil](#), [screenshot](#))

2018 : APT 28

The screenshot shows the assembly view in IDA Pro with several code snippets highlighted by red boxes:

- Call to main.killProcess:** A red box highlights the instruction `call main.killProcess`. A red arrow points from this box to another red box containing the text "Enumerate & Kill the process".
- Loc_1403A62D9:** A red box highlights the label `loc_1403A62D9`.
- Loc_1403A6209:** A red box highlights the label `loc_1403A6209`.
- Loc_1403A6300:** A red box highlights the label `loc_1403A6300`.
- Loc_1403A62B9:** A red box highlights the label `loc_1403A62B9`.

Next, it uses an open source re-written version of `PSUITL` in golang to kill the process, here it is Firefox. You can find the project [here](#).

2024 : APT 36

2022 : Arid Viper

Implants alignment with the common go-malware landscape.

As per my analysis, the implant contains a lot of open-source projects used for various purposes, similar to the implants or generic stealers I have analyzed, like the `kbinani` project for screen grabbing, so I would say aligning to the current go-malware scenario, the implant is not too different from the other implants or malware samples out there in the wild.

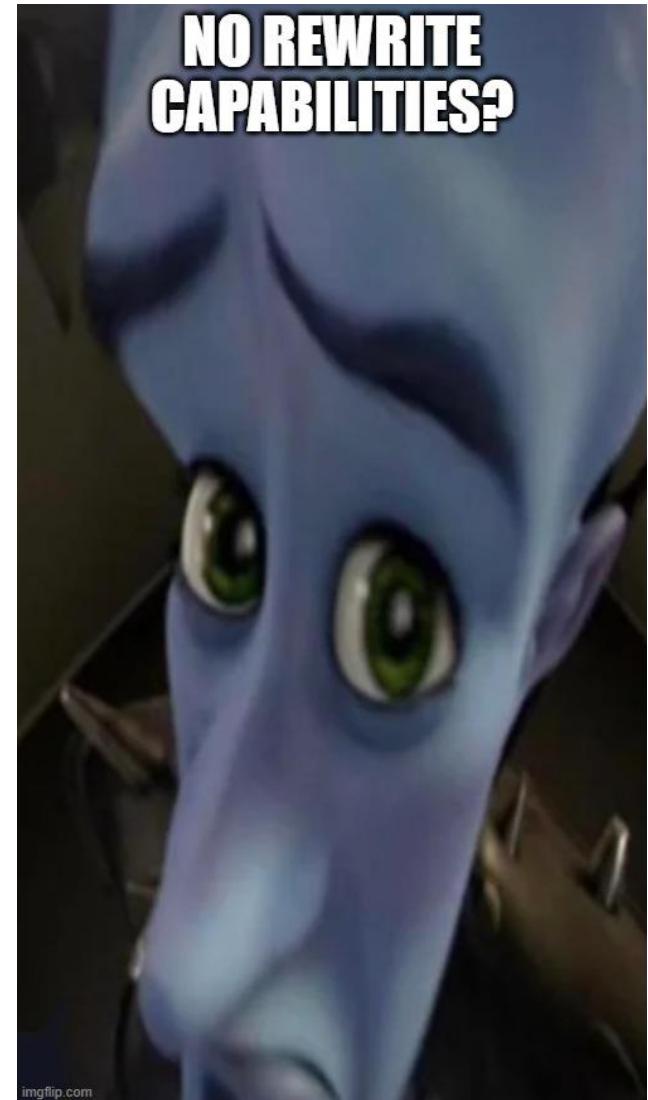
2024 · Patchwork



CASE STUDY : RISE OF TREND

Some common used projects and libraries:

- go-ole.
- os-exec.
- syscall.
- [clipboard.](#)
- crypto
- go-ps.
- debug/pe.
- debug/elf.



imgflip.com



CASE STUDY : RISE OF TREND

Developments in last 5 years ?

- Too many Open-Source Golang Malware projects in GitHub.
- Open-Source Command and Control Frameworks projects being used in campaigns.
- Craze among malicious software developers.
- Easily tweakable.



CASE STUDY : RISE OF TREND

Golang Ransomware:

- [target111/go-crypt](#) - Golang ransomware
- [lucasduete/ransomware-go](#) - Ransomware implementation in Go
- [kvasirlabs/pirategopher](#): Ransomware written in Go w/ a variety of payload options and a easy-to-deploy C2 server ([github.com](#))
- [The-Mario/Mario-Ransomware](#): Ransomware ([github.com](#))
- [Soldie/ransomware](#) ([github.com](#))
- [0xh4di/ransomware](#) ([github.com](#))
- [andyjsmith/Bad-Gopher](#): POC ransomware in Golang ([github.com](#))
- [qureshijey1/ransomware-project](#) ([github.com](#))
- [mauri870/ransomware](#): A POC Windows crypto-ransomware (Academic). Now Ransom:Win32/MauriCrypt.MK!MTB ([github.com](#))
- [wille/cry](#): Cross platform PoC ransomware written in Go ([github.com](#))
- [ejserna/Ransomware](#): Crypto Ransomware made with: - Go for encryption and decryption - PHP/MySQL for saving and retrieving keys. ([github.com](#))
- [gustavohenrique/ransomware](#): A ransomware implementation just for educational purpose ([github.com](#))
- [TheCreeper/UselessLocker](#): Randomware-like sample that can be easily modified and used. For educational purposes. ([github.com](#))

The screenshot shows five GitHub repository cards for Golang malware projects, all updated within the last year. 1. **redcode-labs/Coldfire**: A Go-based malware development library with 935 stars. 2. **EgeBalci/EGESPLOIT**: A Go library for malware development with 338 stars. 3. **omaidf/go-malware**: Examples of Golang viruses with 94 stars. 4. **D3Ext/maldev**: A Go library for malware development with 323 stars. 5. **redcode-labs/neurax**: A framework for constructing self-spreading binaries with 999 stars.

Too many open-source Golang malware projects.



CASE STUDY : RISE OF TREND

discord-c2 Public

Watch 3

main 1 Branch 0 Tags

Go to file Add file Code

William Moody Fixed 'resp undefined' error c46501e · 2 years ago 6 Commits

client.go Fixed 'resp undefined' error 2 years ago

go.mod work at ohme 2 years ago

go.sum work at ohme 2 years ago

Used by Transparent Tribe / APT 36

- [BishopFox/sliver](#) - Adversary Emulation Framework
- [sensepost/godoh](#) - godoh - A DNS-over-HTTPS C2
- [neox41/go-smbshell](#) - Proof of concept SMB C2 using named pipes in Golang
- [audibleblink/gorsh](#) - A Golang Reverse Shell w/ a Tmux-driven pseudo-C2 Interface
- [degenerat3/meteor](#) - A cross-platform C2/teamserver supporting multiple transport protocols, written in Go
- [m00zh33/golang_c2](#) - Boilerplate C2 written in Go for red teams
- [jm33-m0/emp3r0r](#): Linux/Windows post-exploitation framework made by linux user (github.com)
- [lu4p/ToRat](#): ToRat is a Remote Administration tool written in Go using Tor as a transport mechanism and RPC for communication (github.com)
- [sensepost/godoh](#): godoh - A DNS-over-HTTPS C2 (github.com)
- [KCarreto/paragon](#): Red Team engagement platform with the goal of unifying offensive tools behind a simple UI (github.com)
- [averagesecurityguy/c2](#): A simple, extensible C&C beaconing system. (github.com)
- [mawiwillme/Smeagol](#): C2 framework to rule all frameworks (github.com)

Used by many APT groups and ransomware groups.



CASE STUDY : RISE OF TREND

 Other telebackdoor
NO AVATAR

I'm leaking a backdoor for Windows, I wrote in **Golang** there is only the source code of the bot, the rest is created through build.bat, in case of any incomprehensible problem, contact PM to help. Reference: <https://fex.net/ru/s/azpryf>

Homeless
Subject
23.10.2024
#backdoor
Replies: 0
Section: [Software](#)

 Running Python Code Through **Golang**: Obfuscation and Encryption
NO AVATAR

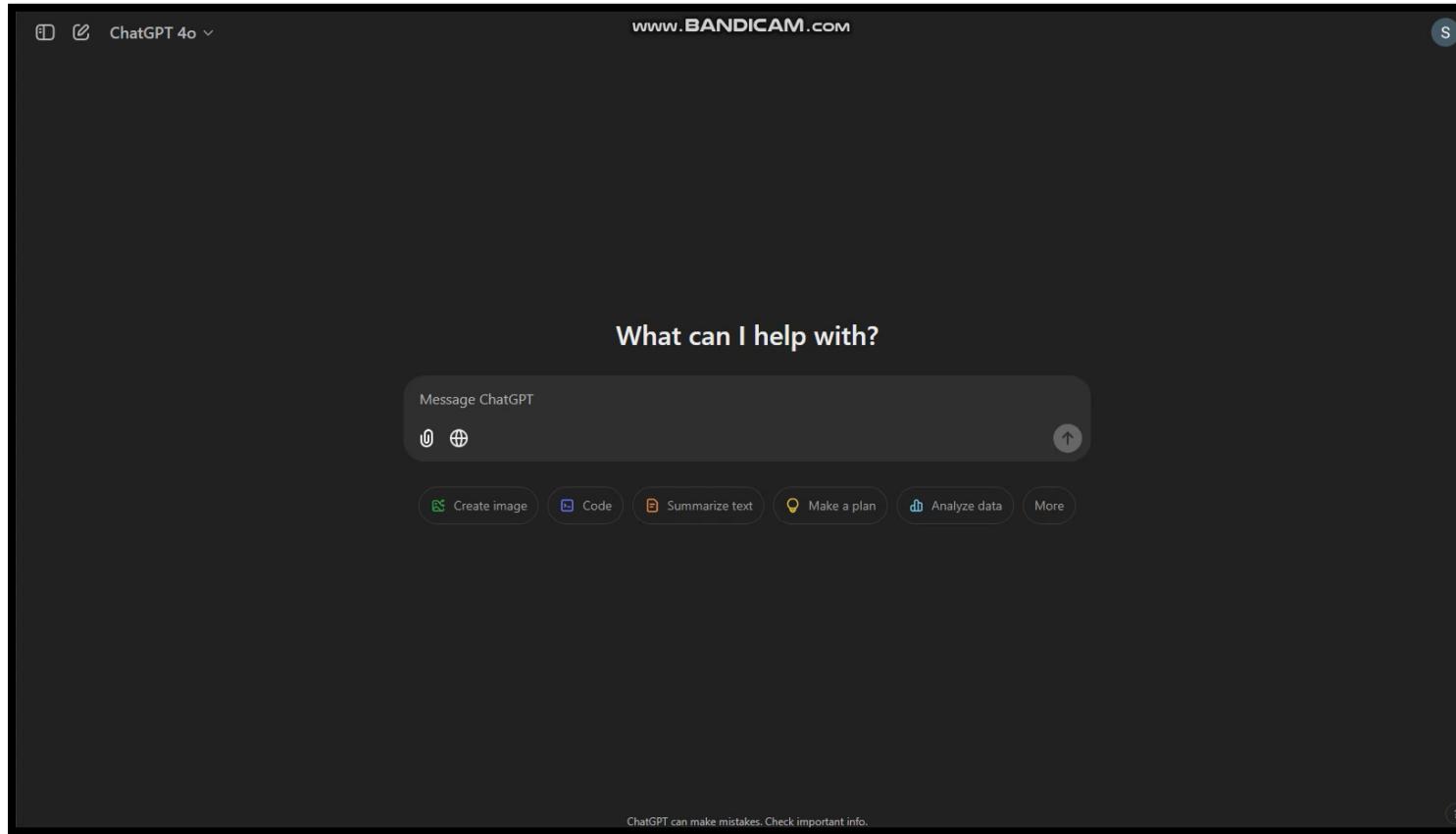
Has anyone seen this repository on GitHub? Seems to be very useful for obfuscated execution, **golang -> python** <https://github.com/cpunion/go-python> Wrote a small POC code package main import (gp "github.com/cpunion/go-python" // Import the go-python wrapper for...

HSVector
Subject
02.11.2024
Replies: 0
Section: [MALWARE: malware, crypto, injects, 0/1day exp](#)

Rise in Golang oriented topics in XSS Forum.



CASE STUDY : RISE OF TREND



Easily Tweakable & easy to write

[Even anyone with an access to GPT could do it]



CASE STUDY : FALL OF TREND

CLAIMS FOR “FALL OF GOLANG MALWARE”

- Adaptation of Malware Research Tools and Strategies.
- Operator Errors Facilitating Proactive Defense.
 1. Unique Indicators and Artifacts.
 2. Abuse of Open-Source Repositories.



Claim: Adaptation of Malware Research Tools and Strategies.

“The rise of Golang malware prompted the development of specialized tools and techniques, including reverse engineering plugins and detection rules tailored to Golang binaries, which have significantly improved detection rates.”





The screenshot shows the GitHub repository for gootools. It has 12 watchers, 35 forks, and 309 stars. The master branch has 14 commits from felberj. The repository description states it's a plugin for Ghidra to assist reversing Golang binaries. Tags include golang, reverse, and ghidra. Files listed include .github/ISSUE_TEMPLATE, data, and ghidra_scripts.

AlphaGolang | A Step-by-Step Go Malware Reversing Methodology for IDA Pro

👤 JUAN ANDRÉS GUERRERO-SAADE / 📅 OCTOBER 21, 2021

Genetic Malware Analysis for Golang



Written by **Intezer** - 20 November 2019

GoUtils2.0

master GoUtils2.0

History Find file Code

Update README.md Egor Zaytsev authored 7 years ago

a8a1677d

Name	Last commit	Last update
GO_Utils	Initial	8 years ago
.gitignore	Initial	8 years ago
README.md	Update README.md	7 years ago
go_entry.py	Initial	8 years ago

Research on dealing with
Golang Malware



CASE STUDY : FALL OF TREND

SEQRITE

Quick Heal

2024-11-04 17:36	a62cef6dc0e8e55806184...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 17:26	3bd7660e77cbac4a9c8e...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 17:11	e73d0a5c7b43c9109708...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 16:26	b0d15ae15dde91eba49...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 16:16	35808f69f5f76ddd48c26...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 16:16	e57cfb75a9ebe94ae0d63...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:56	85c0c57eed2a9c08b0091...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:56	e1e42011155b6a06f0620...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:46	63e389a3d5251cbeaaab...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:26	f0b064eb06f164b0d61e6...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:22	4b159ac65cfa47d511af5...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:16	a7e8c3992382bc855aaf2...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:16	d815fe9c20106fa19135d...	elf	Kaiji	elf kaiji	abuse_ch	
2024-11-04 15:16	0b3a333c1f4c6802a71f9...	elf	Kaiji	elf kaiji	abuse_ch	

Community based YARA
Rules on Golang Malware.
Huge thanks to Abuse.ch!

2024-11-07 12:44	181149cbea8570220cfb6...	exe	GOStealer	exe GOStealer	JAMESWT_MHT	
2024-11-07 02:19	f2ecabc649b7db40d38a8...	exe		exe	SecuriteInfoCom	
2024-11-06 15:41	e78fc7300dea3f82b9fb7...	exe	GOBackdoor	exe GOBackdoor	abuse_ch	

Claim: Operator Errors Facilitating Proactive Defense.

“The quick adoption of Golang Malware among developers has also led to mistakes like giving away too much artefacts along with the malware implants along with which TAs heavily abuse open source repositories, which already are monitored by Anti-Malware Vendors ”



43 / 73
Community Score

43/73 security vendors flagged this file as malicious

d3aadb... PayloadLoader.exe

Size: 20.43 MB | Last Analysis Date: 3 days ago | EXE

peexe checks-user-input detect-debug-environment idle 64bits spreader

Go build ID: "dyTueqWboJUvMxiI3vKE/jyApOq0ni3Zf0vxx-mwq/oMx9oDAgxMj3jW4x-a3W/4c-BObbGpuuSVMhmhkP0"

Example 1:

Go-build ID Extracted leads to more payloads part of a simple Sliver C2 campaign.

content:476f206275696c642049443a202264795475657157626f4a55764d78694933764b452f6a7941704f67306e69335a66

Smart search Sort by Export Tools Help

THREATS IOCS 2 REPORTS & ANALYSIS RULES GRAPHS COMMENTS

Filters

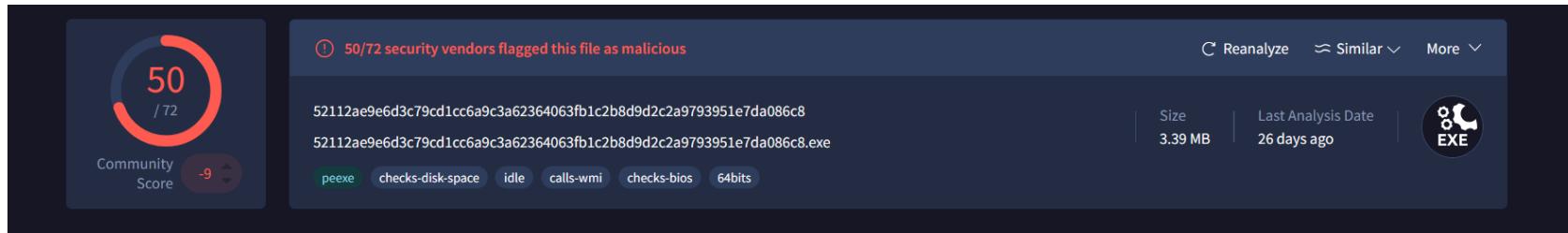
IoC type: Files

GTI Verdict: Benign, Malicious, Suspicious, Undetected

Examples: Have 5 or more detections, Have distribution vectors

90 days

	Summary - 2 Files	Associations	Detections	First seen	Last seen	Submitters
d3aadb... PayloadLoader.exe	sliver	43 / 73	2024-11-17 21:23:41	2024-11-18 11:14:31	2	20.43 MB
2a42826a798d51ba5... PayloadLoader.exe	sliver	37 / 73	2024-11-18 02:49:17	2024-11-18 14:45:27	2	20.43 MB



Example 2:

Recently discovered unknown Infostealer by hunting specific libraries using VT Hunt.

A screenshot of a file browser or debugger interface showing a list of functions in a library named 'torjanPrefix'. One function, 'torjanPrefix_pkg_GetFileContent_func1', is highlighted with a blue selection bar. The list includes other functions like 'torjanPrefix_pkg_GetRAM', 'torjanPrefix_pkg_init_0', etc.

```
torjanPrefix
  pkg
    torjanPrefix_pkg_GetRAM
    torjanPrefix_pkg_init_0
    torjanPrefix_pkg_getRandKey
    torjanPrefix_pkg_customCrypt
    torjanPrefix_pkg_base64Decode
    torjanPrefix_pkg_GetFileContent
    torjanPrefix_pkg_GetFileContent_func1
    torjanPrefix_pkg_getEnv
    torjanPrefix_pkg_traverseDir
    torjanPrefix_pkg_traverseDir_func1
    torjanPrefix_pkg_getFileName
    torjanPrefix_pkg_collectPhoneNumber
    torjanPrefix_pkg_collectBrowserAccounts
    torjanPrefix_pkg_collectFireFoxBrowserAcco
    torjanPrefix_pkg_collectCodepage
    torjanPrefix_pkg_GetRandFileName
    torjanPrefix_pkg_CopyFile2Target
    torjanPrefix_pkg_getAllCryptStr
    torjanPrefix_pkg_CollectSensitiveFileAndSen
    torjanPrefix_pkg_createSocket
    torjanPrefix_pkg_parseCustomDomainName
    torjanPrefix_pkg_ResolveAndConnect
    torjanPrefix_pkg_parseIPPort
    torjanPrefix_pkg_connectIPPool
    torjanPrefix_pkg_packMsg
```

Example 2:

The Go-Build ID is present
inside the malware binary.



```
Go build ID: "IdAGiyd8kv8EskYTKDSe/rwQa9_JAstcSxt8R30Jy/NKkEZtFC-Hl9DyYgKSz1/46pcxpPXq_eLuOQtNGwq"
```

Example 2:

The Go-Build ID leads to other samples ITW.



The screenshot shows a dark-themed user interface for threat intelligence analysis. At the top, there's a search bar with the query "content:476f206275696c642049443a202249644147697964386b763845536b59544b4453652f72775161395f4a417374635". Below the search bar, there are tabs: THREATS, IOCS (with a badge of 8), REPORTS & ANALYSIS, RULES, GRAPHS, and COMMENTS. The IOCS tab is currently selected. On the left, there's a sidebar with "Filters" and "IoC type" set to "Files". It also includes sections for "GTI Verdict" (Benign, Malicious, Suspicious, Undetected) and "Examples" (checkboxes for Have 5 or more detections, Have distribution vectors, Have threat network infrastructure, Have sandbox detonation report, Have community comments). The main area displays a table titled "90 days" with three rows of data. Each row represents a file sample with details like SHA256 hash, file path, detection count, first seen date, last seen date, submitter count, and file size. A blue callout bubble with a magnifying glass icon is positioned over the third row.

	Summary - 8 Files	Associations	Detections	First seen	Last seen	Submitters	
52112ae9e6d3c79cd...	-	50 / 72	2024-08-13 08:26:14	2024-09-17 21:16:28	6	3.39 MB	
bdcf770bb525cdc9...	-	38 / 73	2024-08-16 00:56:58	2024-08-20 18:44:56	2	3.39 MB	
44a99fb49a8fd5d30...	-	36 / 73	2024-08-30 00:29:39	2024-09-17 16:11:57	2	3.39 MB	

CASE STUDY : FALL OF TREND

content: "zavla/dpapi"

Smart search

IoC type: Files (20)

GTI Verdict: Benign, Malicious, Suspicious, Undetected

Examples: Have 5 or more detections (checked), Have distribution vectors, Have threat network infrastructure, Have sandbox detonation report, Have community comments, Seen in the last month

	Summary - 20 Files	Associations	Detections	First seen	Last seen	Submitters	
5a2a8b6c6db9b364a8...	@ /home/petik/...	TorLoader	44 / 73	2024-11-20 01:30:17	2024-11-20 01:30:17	1	EXE 5.84 MB
99e88edaa6cd3c45a1...	@ C:\Program F...	TorLoader	43 / 72	2024-11-20 01:33:55	2024-11-20 01:33:55	1	EXE 5.84 MB
c196651180b2a2b581...	@ C:\Program F...	TorLoader	41 / 72	2024-11-20 01:39:12	2024-11-20 01:39:12	1	EXE 5.84 MB
fbf011323dc02991b9...	@ /home/petik/...	TorLoader	43 / 72	2024-11-20 01:49:07	2024-11-20 01:49:07	1	EXE 5.84 MB
bcb78710ddb6de48ec...	@ C:\Program F...	TorLoader	44 / 73	2024-11-20 01:57:06	2024-11-20 01:57:06	1	EXE 5.84 MB

Some overused libraries,
in Golang malware.

CASE STUDY : FALL OF TREND

SEQRITE

Quick Heal

The screenshot shows a list of detected files from a malware analysis tool. The files are:

- 5360c285015a044325... (EXE, CobaltStrike, 1.67 MB)
- 3c0ae3dd26dd1ae362... (EXE, CobaltStrike, 1.67 MB)
- 5fb3febc11d34272d2... (EXE, MerlinAgent, 11.89 MB)
- 072c62923b3aeb0bfa... (ELF, ScareCrow, 9.77 MB)
- d450853152de0171ec... (EXE, Doge-MemX.exe, 3 overused libraries)

The interface includes a search bar at the top left, a toolbar with various icons, and a bottom navigation bar.

Some overused libraries,
in Golang malware.



CASE STUDY : PSEUDO RISE

CLAIMS/HYPOTHESES FOR “PSEUDO-RISE”

- Abuse of Obfuscation Tools.
- Looking into Obfuscation Tools.
- Increased Detection Rates.



Claim: Abuse of Obfuscation Tools.

“The quick adoption of Golang Malware among developers has also led to mistakes like giving away too much artefacts along with the malware implants lead TAs shift to obfuscators and anti-analysis measures such as Gobfuscator, Garble and many such tools giving them a sense of pseudo-rise.”



Obfuscation

When it comes to obfuscation, various tactics make Blackrota difficult to analyze and detect. For one, the malware uses [gobfuscate](#), an open-source tool for Go code, to obfuscate the source code before compiling. It hides various elements of Go source code with random character substitutions – including the package names, global variable names, function names, type names and method names.

"With thousands of random string-named functions and a large number of randomly-named data types, methods and global variables, we could not be sure what third-party Go packages were used inside the sample, making the reverse-analysis almost impossible to move forward," said researchers.

MetaStealer Obfuscated Go Executable

The main executable in MetaStealer bundles is an Intel x86 Mach-O containing compiled and heavily obfuscated Go source code. The Go Build ID has been stripped and function names obfuscated. The obfuscation method bears similarity to that used in obfuscated Sliver and Poseidon malware binaries, and may be a product of the [garble obfuscator](#) or similar.

GoRed analysis

Before we proceed to analyzing the current version of **GoRed**, we will provide a retrospective analysis of its evolution.

Versions we found

All the versions we found are shown in the table below.

Version	Description
0.0.1	<ul style="list-style-type: none">Assumed to be the first one.Collects information about the victim.Source obfuscated with garble.

RansomHub and Knight compared

Both payloads are written in Go and most variants of each family are obfuscated with [Gobfuscate](#). Only some early versions of Knight are not obfuscated.

The degree of code overlap between the two families is significant, making it very difficult to differentiate between them. In many cases, a determination could only be confirmed by checking the embedded link to the data leak site.

Snake Ransomware (or EKANS Ransomware) is a Golang ransomware which in the past has affected several companies such as Enel and Honda. The MD5 hashing of the analyzed sample is [ED3C05BDE9F0EA0F1321355B03AC42D0](#) ⁵. This sample in particular is obfuscated with [Gobfuscate](#) ⁴, an open source obfuscation project available on Github.

Claim: Looking into Obfuscation tools.

“We will look into some of the artefacts of these obfuscation tools like Gobfuscator and Garble.”

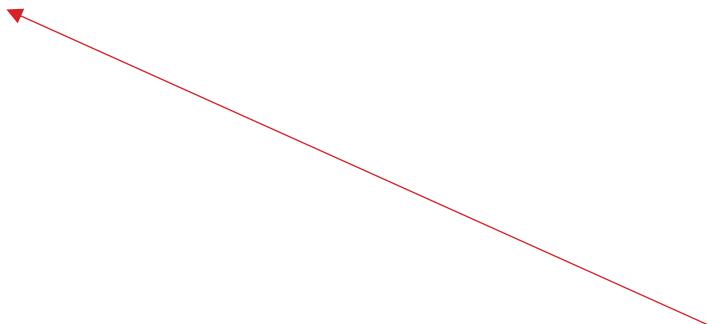


README BSD-2-Clause license

gobfuscate

When you compile a Go binary, it contains a lot of information about your source code: field names, strings, package paths, etc. If you want to ship a binary without leaking this kind of information, what are you to do?

With gobfuscate, you can compile a Go binary from obfuscated source code. This makes a lot of information difficult or impossible to decipher from the binary.



We will lurk into some of the interesting artefacts of this tool and what happens behind the scenes.



About : Some interesting features.

- **Winhide** : This feature could help a binary to run without a GUI or Window, can let the binary to slowly run in the background.
- **Obfuscate Package Names**: This feature helps the final Go binary to obfuscate the package names excluding the CGO libraries.
- **Obfuscate Strings**: This feature helps the final Go binary to be generated with obfuscated string names excluding the CGO with a random XOR mask.
- **Obfuscate Symbols** : This feature helps the final Go binary to be generated with obfuscated string names by directory walking and AST to find top level methods, functions and hashing it with a random seed provided by the user.



The screenshot shows the GitHub README for the `garble` project. It includes sections for installation (`go install mvdan.cc/garble@latest`), basic usage (`garble build [build flags] [packages]`), and advanced features like testing and de-obfuscation. A note at the bottom suggests using `go install mvdan.cc/garble@master` for the latest development version.

```
garble

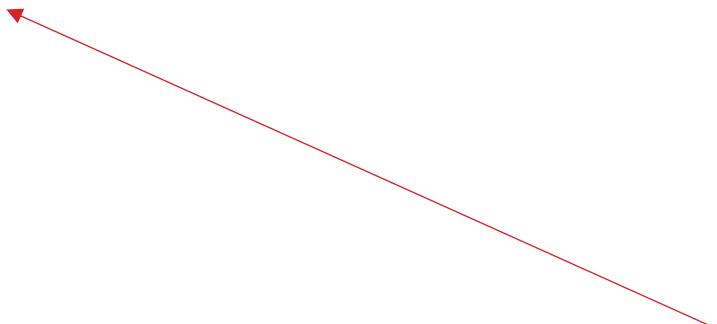
go install mvdan.cc/garble@latest

Obfuscate Go code by wrapping the Go toolchain. Requires Go 1.23 or later.

garble build [build flags] [packages]

The tool also supports garble test to run tests with obfuscated code, garble run to obfuscate and execute simple programs, and garble reverse to de-obfuscate text such as stack traces. Run garble -h to see all available commands and flags.

You can also use go install mvdan.cc/garble@master to install the latest development version.
```



We will lurk into some of the interesting artefacts of this tool and what happens behind the scenes.

About : Some interesting features.

- **Control Flow Obfuscation** : This feature adds bunch of junk code in the obfuscated binary using Garble.
- **Obfuscate Package Names**: This feature helps the final Go binary to obfuscate the package names excluding the CGO libraries.
- **Obfuscate Strings**: This feature helps the final Go binary to be generated with obfuscated string names excluding the CGO with a random XOR mask and much more like ADD, NOR etc.
- **Obfuscate Symbols** : This feature helps the final Go binary to be generated with obfuscated string names by directory walking and AST to find top level methods, functions and hashing it with a random seed provided by the user.
- **Go-Toolchain Integration** : Hooks into the Go-Toolchain seamlessly using –toolexec , ensuring compatibility with existing build pipeline and tools.
- **Anti-Analysis** : Garble employs bunch of anti-analysis techniques which including removing various headers which contain important info.

Claim: Increased Detection Rate.

“The quick adoption of these obfuscators and packers by threat actors and groups have let anti-malware vendors to heavily flag and block these and leading to increased detection rates.”



Malwarebytes

The screenshot shows a forum post on the Malwarebytes website. The post is titled "False positives of obfuscated Go binaries - Malware.AI.1038955078". It was posted by a user named "gophers" on January 2 in the "File Detections" category. The post includes a sample Go code snippet:

```
package main

import "fmt"

func main() {
    fmt.Println("hello")
}
```

Below the code, there is a note: "Build using garble (from master) from the above url and Go 1.21.5".



CONCLUSION



- Detections and blocking on both malicious and non-malicious binaries.
- Threat Actors have taken an alternative route by using different open source packers , obfuscators or their own custom ones.
- A decline in Golang malware scenario as now community support for research against these obfuscators and Golang malware in general is at all time high.
- Peak enthusiasm amongst malware developers to learn Golang.



TOOL-DEMO : GO-PEEP



www.BANDICAM.COM

Go-Peep

Analyze and extract artifacts from Go binaries with ease.

Upload Go Binary

Drag & Drop or Click to Select File

Details Libraries Strings Summary External Scan

> Binary Details

Go Build ID: 0ddYH5b9Vah-bfG9HOJK/KnjeQjm3gZEvs_46XMPw/Mi8f4C_DCwUccVrKRcGI/EBF6-cQjOT7qNH_s60qp
Go Compiler Version: go1.23.2
Binary Type: PE (Windows)
Project Path: N/A
Packed using UPX: No

Export Report



You can find the demo [here](#).



> Artefacts

Binary is not obfuscated

Export Report

> Artefacts

● Binary is obfuscated (No PCLN table, No main function, Has opcodes)

⚠ Warning: Sliver C2 implant detected!

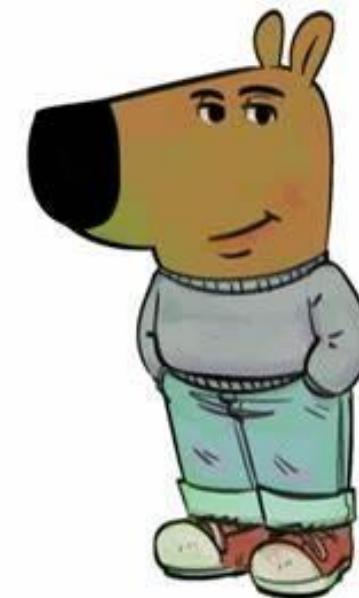


```
Interesting Path: C:/Program Files/Go/src/fmt/format.go
```

```
Interesting Path: C:/Users/Pseudo-user/Desktop/Go-Peep/binary/main.go
```



Any Questions?





Thank You

Innovate. Simplify. Secure.