GitHub - SaturnsVoid/GoBot2: Second Version of The GoBot Botnet, But more advanced.

By SaturnsVoid

Archived: 2026-04-05 21:55:14 UTC

New project: https://github.com/SaturnsVoid/Project-Whis

After seeing another users Go based botnet i wanted to do more work on my GoBot, But i ended up building something a bit more. There is issues with this but it more of a advanced PoC.... I am not a good coder but i was able to make this buy doing some basic reading online. There was more i wanted to do with this project but i stopped, I am getting out of making Malware and virus's... I am going to move on to more legitimet things. Though i will be posting some of my old projects on my Github, and most of witch are malevolent i am putting them here to make it simpler for the 'good guys' to fight them and there kin.

C&C Features:

- Written in Go
- Cross-Platform
- SQL Database for Information
- Secure Login System
- Hard-Coded Login System
- Simple to use HTML & CSS C&C
- Console Based C&C
- Tight Security (No PHP!)
- Encoded and Obfuscated Data
- HTTPS or HTTP
- Single, Selected, All Command Issuing
- User-Agent Detection
- More

Bot Features

- Safe Error Handling
- Have Unlimited Panels
- Encoding and Obfuscation
- Use HTTPS or HTTP
- Old (>24Hr) Command Handling (Dont run commands that are old!)
- Run PowerShell Scripts (Via URL, Parameters Accepted)
- Advanced Torrent Seeder (uTorrent, BitTorrent Auto Download the client and runs hidden if needed)
- Drive Spreader (with Name list)
- Dropbox Spreader (with Name list)
- Google Drive Spreader (with Name list)
- OneDrive Spreader (with Name list)
- Advanced Keylogger (Handles all keys, Window Titles, Clipboard, AutoStart, +more)
- System Information (IP, WiFi, User, AV, IPConfig, CPU, GPU, SysInfo, Installed Software, .NET Framework, Refresher)
- Screen Capture (Compression, Timed Capture, +more)
- Download and Run (MD5 Hash Check, URL or Base64, Parameters, UAC Bypass, Zone Remover)

- DDoS Methods (Threaded /w Interval, HTTPGet, TCPFlood, UDPFlood, Slowloris, HULK, TLSFlood, Bandwidth Drain, GoldenEye, Ace)
- Bot Update (MD5 Hash Check, Admin, Zone Remover)
- UPnP (Open TCP/UDP Ports)
- Web-Server (Auto-UPnP port 80, Add/Edit Unlimited Pages)
- Add Programs to Windows Firewall
- HOST File Editor (Backup and Restore, Replace on Run, DNS Flusher)
- Remote CMD
- Detect Admin Rights
- Bot ID Generation (Never the same)
- Advanced Anti-Virus Bypass (Random Memory Allocation, Func HOP, Delays, Runtime Load DLLS /w Obf, Random Connection Times, + more)
- Advanced Anti-Debug (isDebuggerPresent, Proc Detection, IP Organization Detection, File Name Detection, Reaction System)
- Single Instance System
- Reverse HTTP Proxy (Conf. Port, backend Servers)
- Active Defense (Active Registry Defense, Active File Defense, Active WatchDog + more) Doesn't want to be killed.
- UAC Bypass (Work all versions and current version of Windows 10 Pro 64Bit)
- Advanced Install System (Dynamic Registry Keys, Dynamic File Names, Retain Admin Rights, Campaign Targeting (Only install in allowed Country's), Zone Remover, Adds self to Firewall)
- Uninstall System (Removes all Traces)
- Scripter (Batch, HTML, VBS, PS)
- Run Shellcode (ThreadExecute)
- Power Options (Shutdown, Restart, Logoff)
- Startup Error Message
- MessageBox (Returns Reply)
- Open Website (Visible/Hidden)
- Change Homepage
- Change Background (URL or Base64)
- Run .exe (UAC Bypass optimal)
- Kill Self
- Check if Proc is Running
- Hide Process /w Active Mode
- Disable/Enable (TaskManger, RedEdit, Command Prompt)
- File Dropper (Place evedence on pc with no traces where it came from /w dir selection)

Some Info about the C&C

The C&C is a program, You can compile it for Windows, Linux, Mac systems. Its a self-running web-server that handles all connections on the selected port in the settings. it will serve the HTLM C&C to a connector if you allow it and it saves data about account, bots and commands as a SQL database and bots files (screenshots, keylogs, ect) as file under the bots own "Profile"
You can control the botnet from the program(more secure) or control it from the HTML C&C. The C&C's program is extremely stable, Go based servers are know for handling millions or requests at once without fail, just make sure you have a good connection.
The C&C has a build in hard-coded login (kinda like a Backdoor) you can use if you 'forgot' the account login. the C&C can have any number of accounts.
With it being a self-contained program this removes the issue of SQLi attacks on the C&C so its more SECURE.
The C&C can also run inside a Tor Hidden service if configured right and the client (bot) can connect to it using a onion.to or onion.cab forwarder if needed. Tor can also be used by the bot via a SOCKS proxy... Simple to do, Google it.

How to Build and Use

Bot Settings are located in "Variables.go" Server Setting are located in "Server.go"

Compile GoBot.go with correct settings, Make a MySQL Database and import db file, Compile Server.go with correct settings

- go build -o GoBot.exe -ldflags "-H windowsgui" "C:\GoBot2\GoBot.go"
- go build -0 Server.exe "C:\GoBot2\Console Server\Server.go"

Always compile with '-w -s' ldflags to strip any debug information from the binary.

Included Tools

- Tool for the project (Obfuscator (Char+1) and other crap. w/ source in VB.net)
- Downloader.go (GoLANG Download and Run Example)
- DownloaderWithUAC.go (GoLANG Download and Run Example with UAC Bypass)

Obfuscator

It not really a Obfuscator all it does it move the Char +1 to and A = B, C = D, ect. Simple but it will slow down people wanting to mess with the program and also programs that search for keywords...

Packages Used

- github.com/NebulousLabs/go-upnp
- golang.org/x/sys/windows/registry
- github.com/AllenDang/w32
- github.com/atotto/clipboard
- github.com/StackExchange/wmi

Images

- http://prnt.sc/d67nh5
- http://prnt.sc/d67ogh
- http://prnt.sc/d67oqh
- http://prnt.sc/d6esj3

Credits and Stuff

- https://github.com/decred/gominer
- https://github.com/robvanmieghem/gominer
- https://astaxie.gitbooks.io/build-web-application-with-golang/content/en/04.5.html
- http://www.adlice.com/runpe-hide-code-behind-legit-process/
- http://www.hacking-tutorial.com/tips-and-trick/how-to-enable-remote-desktop-using-command-prompt/
- https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/
- https://mholt.github.io/json-to-go/
- https://sentinelone.com/blogs/anti-vm-tricks/
- http://hackforums.net/showthread.php?tid=5383448
- https://github.com/grafov/hulk
- https://github.com/nhooyr/dos
- https://github.com/marcelki/sockstress
- https://github.com/ammario/ssynflood
- https://github.com/matishsiao/goInfo/blob/master/goInfo_windows.go
- https://github.com/iamacarpet/go-win64api
- https://github.com/oneumyvakin/initme/blob/master/windows.go
- https://github.com/LOLSquad/DDoS-Scripts

- https://github.com/vbooter/DDoS-Scripts
- https://github.com/natefinch/pie
- https://www.windows-commandline.com/enable-remote-desktop-command-line/
- https://www.socketloop.com/tutorials/golang-secure-tls-connection-between-server-and-client
- https://github.com/lextoumbourou/goodhosts
- https://github.com/YinAndYangSecurityAwareness/dreamr-botnet
- https://github.com/mauri870/ransomware
- http://www.devdungeon.com/content/making-tor-http-requests-go
- http://www.darul.io/post/2015-07-22_go-lang-simple-reverse-proxy
- https://github.com/mauri870/powershell-reverse-http
- https://github.com/gh0std4ncer/lizkebab/blob/master/client.c
- https://github.com/EgeBalci/EGESPLOIT
- https://github.com/EgeBalci/HERCULES
- https://github.com/andrewaeva/gobotnet
- https://github.com/SaturnsVoid/GoBot
- https://github.com/petercunha/GoAT
- https://github.com/huin/goupnp
- https://github.com/ytisf/theZoo/tree/master/malwares/Source/Original
- https://github.com/malwares/Remote-Access-Trojan
- https://github.com/kardianos/service
- https://github.com/vova616/screenshot/blob/master/screenshot_windows.go
- http://hackforums.net/showthread.php?tid=5040543
- http://www.calhoun.io/5-useful-ways-to-use-closures-in-go/
- https://blogs.technet.microsoft.com/ilikesql_by_dandyman/2013/03/10/how-to-install-a-msi-file-unattended/
- https://github.com/tadzik/simpleaes
- https://guitmz.com/win32-liora-b/
- https://github.com/rk/go-cron
- https://breakingmalware.com/vulnerabilities/elastic-boundaries-elevating-privileges-by-environment-variables-expansion/
- https://breakingmalware.com/malware/ardbot-a-malware-under-construction/
- https://breakingmalware.com/malware/furtim-malware-avoids-mass-infection/
- https://www.pugetsystems.com/labs/support-software/How-to-disable-Sleep-Mode-or-Hibernation-793/
- https://files.sans.org/summit/Digital_Forensics_and_Incident_Response_Summit_2015/PDFs/TheresSomethingAboutWMIDevon
- https://github.com/jasonlvhit/gocron

Other

Go is a amazing and powerful programming language. If you already haven't, check it out; https://golang.org/

Donations

Please Donate To Bitcoin Address: **1AEbR1utjaYu3SGtBKZCLJMRR5RS7Bp7eE**

News

I just read a article on Bleeping Computer, https://www.bleepingcomputer.com/news/security/backdoored-torrents-infect-movie-tv-fans-with-gobot2-malware/ Seems someone has found a use for this project... I have no involvment with this group or person. I have nothing more to say on this matter.

-Crab Crab
----------Update Log----------

03/15/2017: Intial Upload...

Source: https://github.com/SaturnsVoid/GoBot2