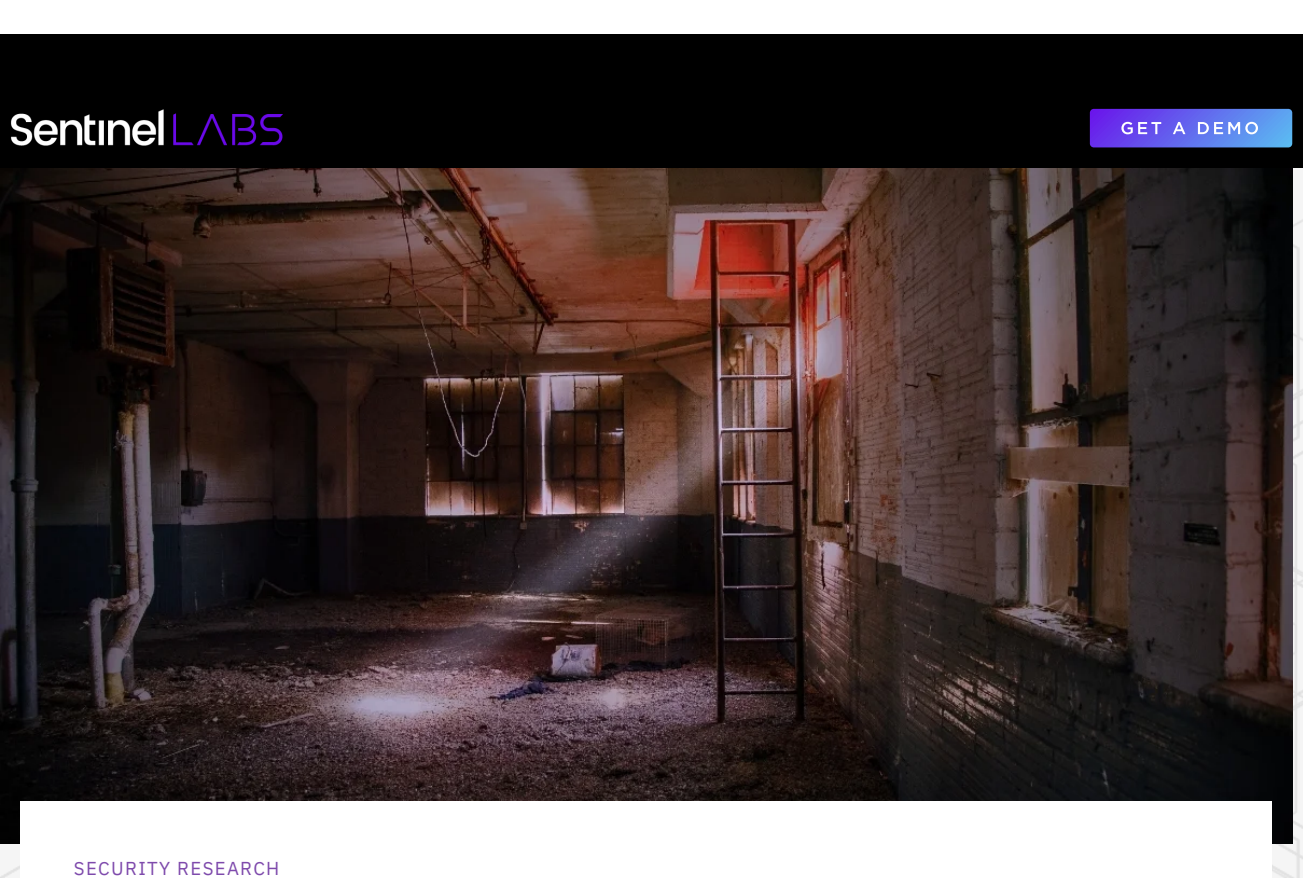


SECURITY RESEARCH

Predator AI | ChatGPT-Powered Infostealer Takes Aim at Cloud Platforms

ALEX DELAMOTTE / NOVEMBER 7, 2023

Executive Summary

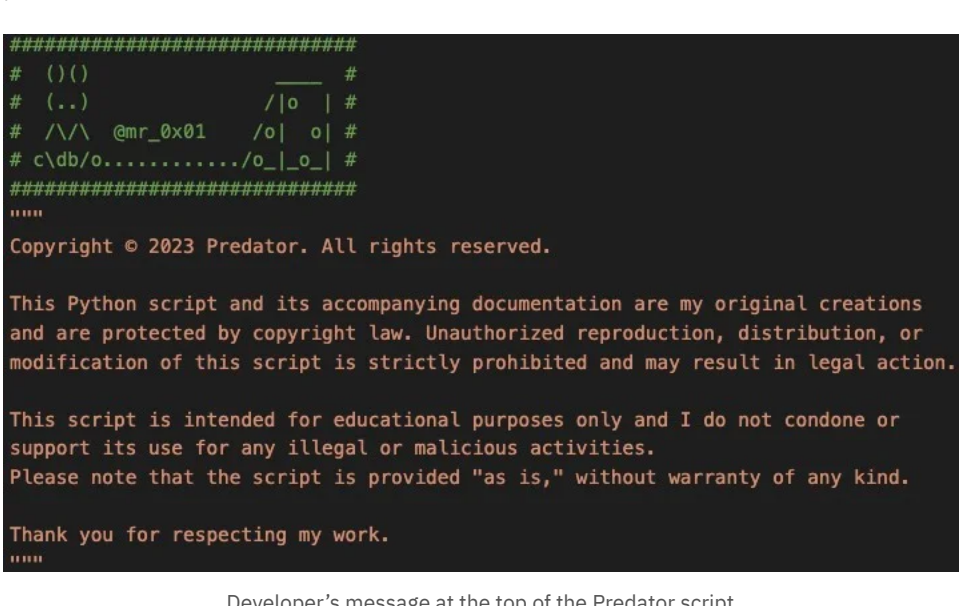
- SentinelLabs has identified a new Python-based infostealer and hacktool called ‘Predator AI’ that is designed to target cloud services.
- The Predator AI developer implemented a ChatGPT-driven class into the Python script, which is designed to make the tool easier to use and to serve as a single text-driven interface between disparate features.
- These advancements are not production ready, but demonstrate that actors can realistically use AI to improve their workflows by automating data enrichment and adding context to scanner results.

Background & Distribution

Predator AI is advertised through Telegram channels related to hacking. The main purpose of Predator is to facilitate web application attacks against various commonly used technologies, including content management systems (CMS) like WordPress, as well as cloud email services like AWS SES. However, Predator is a multi-purpose tool, much like the AlienFox and Legion cloud spamming toolsets. These toolsets share considerable overlap in publicly available code that each repurposes for their brand’s own use, including the use of Androxgh0st and Greenbot modules.

Predator is an actively developed project. In September 2023, a member of the primary Telegram channel inquired about Predator adding a Twilio account checker, to which the developer replied they could deliver in about 2 weeks. In October, the developer posted an update showing the new Twilio checking feature. The version we analyzed has Twilio features, which suggests it is a recent build.

At the top of the script, there is a message from the developer which states that the tool is protected by copyright law. The message also has a disclaimer saying the tool is for educational purposes and the author does not condone any illegal use.


##############################
()()   ____
(..)   /|o |
/\/\ @mr_0x01   /o| o|
c\db/o............/o_|_o_|
##############################
"""
Copyright © 2023 Predator. All rights reserved.
This Python script and its accompanying documentation are my original creations and are protected by copyright law. Unauthorized reproduction, distribution, or modification of this script is strictly prohibited and may result in legal action.
This script is intended for educational purposes only and I do not condone or support its use for any illegal or malicious activities.
Please note that the script is provided "as is," without warranty of any kind.
Thank you for respecting my work.
"""

Developer’s message at the top of the Predator script

Targeting & Technical Details

Predator is a Python application with over 11,000 lines. The application runs entirely through a Tkinter-based graphical user interface (GUI): there is no standalone command line interface (CLI) mode, which distinguishes Predator from many similar tools. The Tkinter approach requires several JSON configuration files.


Predator v0.0.8
Result
Wordpress
Joomla
Prestashop
Drupal
osCommerce
vBulletin
Laravel
Other
SQL
XSS
GIT
Powered By
@mr_0x01
@hofnar05
@Bore3da
@dollifacee
@OBHACKING
@BENUMBZ
Leaked
DBs: 0
SMTP: 0
Twilio: 0
SMS: 0
AWS: 0
Stripe: 0
ClickSend: 0
OneSignal: 0
RazorPay: 0
Tools
Stealer
Logs2Wallet
Generator
Reverse IP
SubFinder
Extractor
NetGun
Grabber
SMTP
Translator
Setup your Settings
Results
SQL: 0
GIT: 0
Config: 0
PHPUnitRCE: 0
com_jce: 0
XSS: 0
CORS: 0
Import
Start
Dark
Threads: 4
Timeout: 30

Predator GUI

The script has 13 global classes defined, which roughly segment the different features.

Class Name	Details
Predator	The largest class. Goes from the beginning to line 7079.
Settings	Only two lines. Sets UpdatesCheck variable to False and Password to “Predator123”.
Utility	Contains calls to Windows commands that get the current window name and to check if the current user is running as an administrator.
PumperSettings	Code that inflates the size of a file.
FakeErrorBuilder	Creates fake error messages that pertain to XSS testing on a Windows system.
StealerBuilder	Builds a configurable infostealer as a Windows Portable Executable (PE).
Translator	Translates the dialog boxes and menu items that are rendered in the GUI version of the application via Python library Tkinter. Supported languages are Arabic, English, Japanese, Russian, and Spanish.
NetGun	Handles web application security scans with options for proxies and custom wordlists.
CTkMessagebox & CTkListbox	Code that renders the graphical user interface (GUI) via Tkinter.
ThemeMaker	Custom color schemes for the GUI.
GPTj	A ChatGPT-enabled class. Queries the OpenAI API.
NetXplorer	Uses Psutil and Subprocesss to query network status and system information.

Predator has features that can be used to attack many popular web services and technologies, including:

Service Provider	Details	Based In
Aimon	SMS marketing	Italy
Amazon Web Services (AWS) Simple Email Service (SES)	Email platform	United States
Aruba	Hosting	Italy
Clickatell	SMS marketing	South Africa, United States
ClickSend	SMS marketing	Australia
Twilio	SMS, Voice, Video communications	United States
Nexmo	Voice & SMS, acquired by Vonage	United States
OneSignal	SMS, Push Notifications	United States, United Kingdom
Openpay	Buy Now, Pay Later; ceased operations in February 2023	Australia
PayPal	Live environment & Sandbox API keys targeted	United States
Plivo	Voice & Messaging	United States
Razorpay	Payment Processor	India
Skebby	SMS Marketing	Italy
Stripe	Payment Processor	United States
Telnyx	Voice, Messaging, Fax	United States
Textlocal	SMS Marketing	United Kingdom
Valueleaf	Marketing	India
XGATE	Marketing & CRM	Hong Kong

Predator’s web application attacks look for common weaknesses, misconfigurations or vulnerabilities in Cross Origin Resource Sharing (CORS), exposed Git configuration, PHPUnit Remote Code Execution (RCE), Structured Query Language (SQL), and Cross-Site Scripting (XSS).

The following technologies are targeted:

- Drupal
- Joomla
- Laravel
- Magento
- OpenCart
- osCommerce
- PrestaShop
- vBulletin
- WordPress


#####RESULT######
PhpUnitRCE = 0
envres = 0
xssres = 0
sqlres = 0
cors_res = 0
gitres = 0
comJceres = 0 # com_jce results
gthosts = 0 # Reverse IP dynamic result
gsubf = 0 #SubFinder
sentCount = 0 #Mailer
ValidSmtps = 0 #SMTP CHecker
BadSmtps = 0 #SMTP CHecker
TotalSmtpsHits = 0 #Laravel Leaked smtps
TotalTwilioHits = 0 #Laravel Leaked Twilio
TotalDBsHits = 0 #Laravel Leaked DBs
TotalAwsHits = 0 #Laravel Leaked AWS
TotalClicksendHits = 0 #Laravel Leaked ClickSend
TotalOsHits = 0 #Laravel Leaked OneSignal
TotalSmsHits = 0 #Laravel Leaked SMS
TotalStripeHits = 0 #Laravel Leaked Stripe
TotalRazorpayHits = 0 #Laravel Leaked RazorPay
TotalpaypalLHits = 0 #Laravel Leaked PP LIVE
TotalpaypalSHits = 0 #Laravel Leaked PP SANDBOX

Variables that hold output from web service scanning features


if "APP_NAME=" in GetConfig.text:
if "APP_ENV=" in GetConfig.text:
self.get_db(GetConfig.text)
self.get_smtp(GetConfig.text)
self.get_aws(GetConfig.text)
self.get_stripe(GetConfig.text)
self.get_razorpay(GetConfig.text)
self.get_skebby(GetConfig.text)
self.get_clickatell(GetConfig.text)
self.get_twilio(GetConfig.text)
self.get_plivo(GetConfig.text)
self.get_aruba(GetConfig.text)
self.get_nexmo(GetConfig.text)
self.get_paypal_sandbox(GetConfig.text)
self.get_paypal_live(GetConfig.text)
self.get_onesignal(GetConfig.text)
self.get_telnyx(GetConfig.text)
self.get_textlocal(GetConfig.text)
self.get_value_leaf(GetConfig.text)
self.get_sms(GetConfig.text)
self.get_openpay(GetConfig.text)
self.get_clicksend(GetConfig.text)
self.get_xgate(GetConfig.text)
self.get_aimon(GetConfig.text)

Laravel environment parsing

Predator AI | The GPTj Class

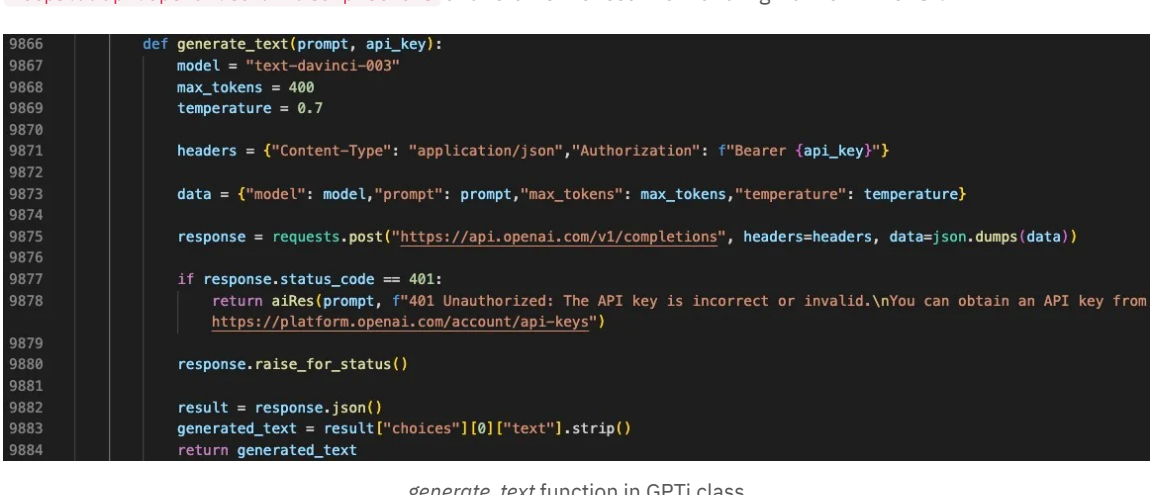
The GPTj class contains the ‘Predator AI’ feature, which is a chat-like text processing interface that connects the user to Predator’s features. The actor designed Predator AI to try to find a local solution first before querying the OpenAI API, which reduces the API consumption.

This class searches the user’s input for strings associated with a known use case centered around one of Predator’s web application and cloud service hacking tools. There are more than 100 cases where Predator handles the data internally or through a free third-party service, such as an IP reputation lookup service. This class contains several partially implemented utilities related to AWS SES and Twilio, as well as utilities to get information about IP addresses and phone numbers.

Predator queries the ChatGPT API only when there is no test case to handle the input. There are several driving functions defined inside this class that handle the activity flow or enable ChatGPT interaction:

generate_text

This function requires two arguments: `prompt` and `api_key`. The function uses the OpenAI model `text-davinci-003` with a maximum token length of 400 and temperature 0.7. The code makes a POST request to `https://api.openai.com/v1/completions` and returns the result for handling via the Tkinter UI.


def generate_text(prompt, api_key):
model = "text-davinci-003"
max_tokens = 400
temperature = 0.7
headers = {"Content-Type": "application/json","Authorization": f"Bearer {api_key}"}
data = {"model": model,"prompt": prompt,"max_tokens": max_tokens,"temperature": temperature}
response = requests.post("https://api.openai.com/v1/completions", headers=headers, data=json.dumps(data))
if response.status_code == 401:
return aiRes(prompt, f"401 Unauthorized: The API key is incorrect or invalid.\nYou can obtain an API key from https://platform.openai.com/account/api-keys")
response.raise_for_status()
result = response.json()
generated_text = result["choices"][0]["text"].strip()
return generated_text

generate_text function in GPTj class

Ai_Backend

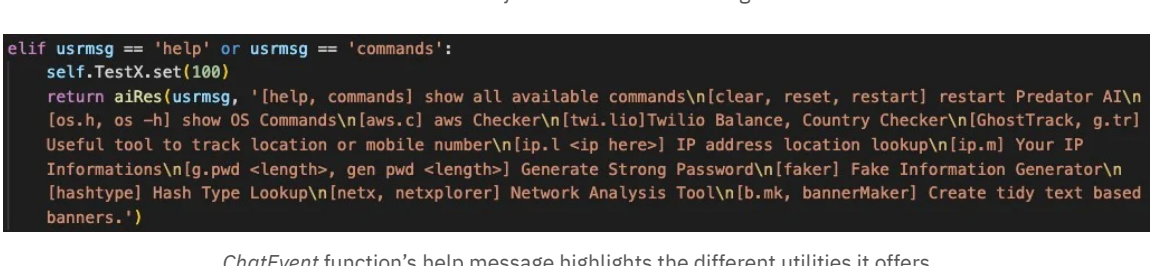
This function takes one argument, `usrMsg`. This code contains the hardcoded OpenAI API key and calls the `generate_text` function on the `usrMsg` object with the API Key. The OpenAI server response is returned.

aiRes

This function takes two arguments, `msg` and `patch`. This function only calls `Ai_Backend` –and OpenAI as a result– when the patch argument is equal to 0, or not given. Predator has 106 references to `aiRes` and each reference has a patch value that should not equal 0. This means the OpenAI functionality is designed to handle edge cases that the script has not natively handled. The function processes whether a patch is present and modifies the UI result based on the length of the response from OpenAI or the patched result.

ChatEvent

This function contains the modular utilities offered by the class. It takes no arguments.


elif usrmsg == 'help' or usrmsg == 'commands':
self.TestX.set(100)
return aiRes(usrmsg, '[help, commands] show all available commands\n[clear, reset, restart] restart Predator AI\n[os.h, os -h] show OS Commands\n[aws.c] aws Checker\n[twi.lio]Twilio Balance, Country Checker\n[GhostTrack, g.tr] Useful tool to track location or mobile number\n[ip.l <ip here>] IP address location lookup\n[ip.m] Your IP Informations\n[g.pwd <length>, gen pwd <length>] Generate Strong Password\n[faker] Fake Information Generator\n[hashtype] Hash Type Lookup\n[netx, netxplorer] Network Analysis Tool\n[b.mk, bannerMaker] Create tidy text based banners.')

ChatEvent function’s help message highlights the different utilities it offers

When the user command is not routed to ChatGPT, several functions handle the request locally or through alternate API calls. We break them down by category.

AWS Features

Though the core utility is present, not all of the following functions are called inside the script, suggesting the developer is still working on these features. This code has significant overlap with AlienFox, Legion, and other earlier iterations of these tools. Based on what is currently in the script, there is no indication that AWS-related data would be sent to the ChatGPT service. Instead, the script parses the input for the presence of `aws.c` and calls the following functions when present.

If these features were fully implemented, the attacker could use them to perform the following when they have valid AWS account credentials:

- Check for all email accounts in an AWS SES environment.
- Check send quotas.
- Create a new account, assign administrative privileges, and delete the old account.

TwilioChecker

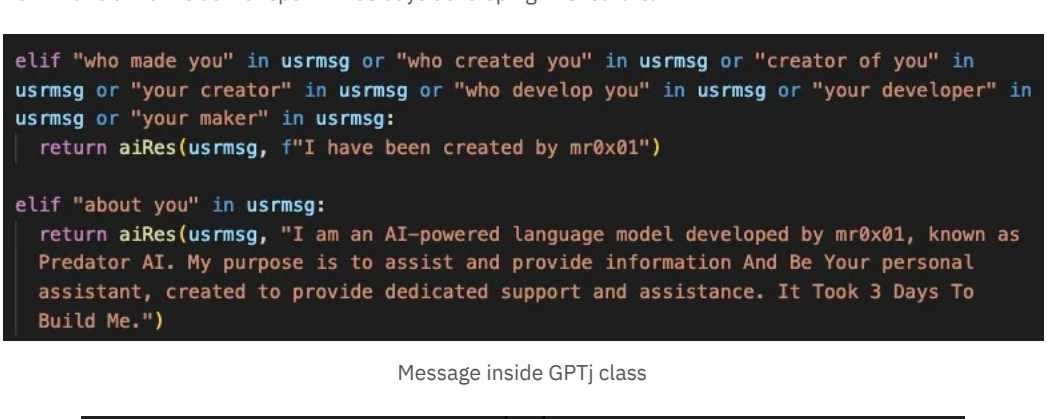
This function queries `https://api.twilio.com/2010-04-01/Accounts.json` with `SID` and token as arguments. If `"message":"Authenticat"` is not in the response, the script parses the response for the fields `status`, `type`, and `balance`. If “status” is not in the response, the script parses the response for balance and currency fields. If status returns as active, the script logs the values of `SID`, `TOKEN`, `TYPE`, `STATUS`, `BALANCE` to the file `Result/TwilioChecker/result.txt`.

GhostTrack

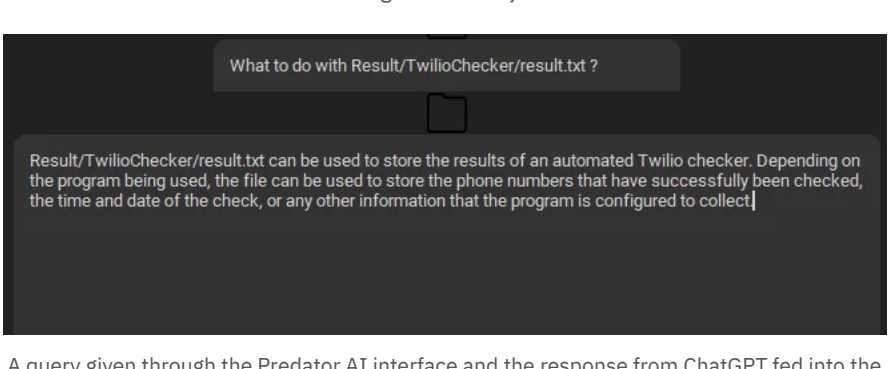
There are several other utilities nested under a function named `GhostTrack`.

- `IP_Track`: Collects information about a given IP address via the `ipwho[.]is` service.
- `phoneGW`: Uses the `phonenumbers` Python module to format input phone numbers in a standard way and check information about the phone number, such as whether it is a landline or mobile number.
- `TrackLu`: Checks one of 23 social media services for a username matching the input argument. The function checks for a 200 status code, which is not effective in the case of private profiles and there are likely many site-specific edge cases.
- `checkIP`: Queries `api.abuseipdb[.]com` to collect information about the given IP address related to abuse metrics, such as an abuse confidence score.

The author included several conditions to handle a user query about the nature of the chat utility, along with a statement that claims the author spent three days developing this feature.


elif "who made you" in usrmsg or "who created you" in usrmsg or "creator of you" in usrmsg or "your creator" in usrmsg or "who develop you" in usrmsg or "your developer" in usrmsg or "your maker" in usrmsg:
return aiRes(usrmsg, f"I have been created by mr0x01")
elif "about you" in usrmsg:
return aiRes(usrmsg, "I am an AI-powered language model developed by mr0x01, known as Predator AI. My purpose is to assist and provide information And Be Your personal assistant, created to provide dedicated support and assistance. It Took 3 Days To Build Me.")

Message inside GPTj class


What to do with Result/TwilioChecker/result.txt ?
Result/TwilioChecker/result.txt can be used to store the results of an automated Twilio checker. Depending on the program being used, the file can be used to store the phone numbers that have successfully been checked, the time and date of the check, or any other information that the program is configured to collect.

A query given through the Predator AI interface and the response from ChatGPT fed into the UI

StealerBuilder

This class contains configuration variables to build an infostealer. On October 16 2023, the project developer posted a video about Predator that shows the Stealer build process. A user asked if the resulting executable is fully undetectable, to which the developer replied, “Of course.”

The stealer can be configured to use Discord or Telegram webhooks for C2. The operator can specify an existing executable to insert the infostealer code into. During testing, we were unable to successfully use this feature as the required configuration files were not available. The features visible in the script we analyze indicate that Predator parses files from a Scripts directory and uses those to build either a Windows Portable Executable (PE) file or a Python script version of the stealer module.


class StealerBuilder(customtkinter.CTkToplevel):
def buildButtonControl_Callback(self) -> None:
"modules" : {
"captureWebcam" : self.captureWebcamVar.get(),
"capturePasswords" : self.capturePasswordsVar.get(),
"captureCookies" : self.captureCookiesVar.get(),
"captureHistory" : self.captureHistoryVar.get(),
"captureAutofills" : self.captureAutofillsVar.get(),
"captureDiscordTokens" : self.captureDiscordTokensVar.get(),
"captureGames" : self.captureGamesVar.get(),
"captureWifiPasswords" : self.captureWifiPasswordsVar.get(),
"captureSystemInfo" : self.captureSystemInfoVar.get(),
"captureScreenshot" : self.captureScreenshotVar.get(),
"captureTelegramSession" : self.captureTelegramVar.get(),
"captureCommonFiles" : self.captureCommonFilesVar.get(),
"captureWallets" : self.captureWalletsVar.get(),
"fakeError" : self.fakeErrorData,
"blockAvSites" : self.blockAvSitesVar.get(),
"discordInjection" : self.discordInjectionVar.get()
},

StealerBuilder configuration variables

Conclusion

The discovery of Predator AI is an entirely expected evolution that has previously been undocumented in the hacktool space. Since the recent wave of AI technologies entered the public domain, security professionals have questioned whether this technology was already aiding threat actors and how it could be used to scale actor operations. There were several projects like BlackMamba that ultimately were more hype than the tool could deliver. Predator AI is a small step forward in this space: the actor is actively working on making a tool that can utilize AI.

While Predator AI is likely somewhat functional, this integration does not substantially increase an attacker’s capability. The feature has not yet been advertised on the actor’s Telegram channel, and there are likely many edge cases that make it unstable and potentially expensive.

Like other cloud service attack tools, organizations can reduce the impacts from these tools by keeping web services patched and up to date, as well as keeping internet access restricted to what is necessary. Use cloud security posture management (CSPM) tools to validate that configurations are secure. Consider dedicated logging and detections for anomalous behaviors on cloud service provider (CSP) resources, such as new user accounts being added and deletion of another user account immediately after.

Indicators of Compromise

SHA-1 Hash

88d40f86eefee5112515b73cce2d2badb7f49ffd – *main.py* Predator Python script

Hardcoded Strings

- “jSDSgnditikunggobloktolol” – hardcoded AWS account name string
- “titid” – hardcoded username in AWS GPT functionality
- “Adminn” – hardcoded username in AWS GPT functionality
- “Predator123” – hardcoded password from the Settings class
- “admainkontolpaslodsajijsd21334#1ejeg2shehhe” – hardcoded password for ‘Kontolz’ user account
- arn:aws:iam::320406895696:user/Kontolz – example ARN for Kontolz user

CLOUD ATTACK SURFACE

ALEX DELAMOTTE

Alex's passion for cybersecurity is humbly rooted in the early aughts, when she declared a vendetta against a computer worm. Over the past decade, Alex has worked with blue, purple, and red teams serving companies in the technology, financial, pharmaceuticals, and telecom sectors and she has shared research with several ISACs. Alex enjoys researching the intersection of cybercrime and state-sponsored activity. She relentlessly questions why actors pivot to a new technique or attack surface. In her spare time, she can be found DJing or servicing her music arcade games.