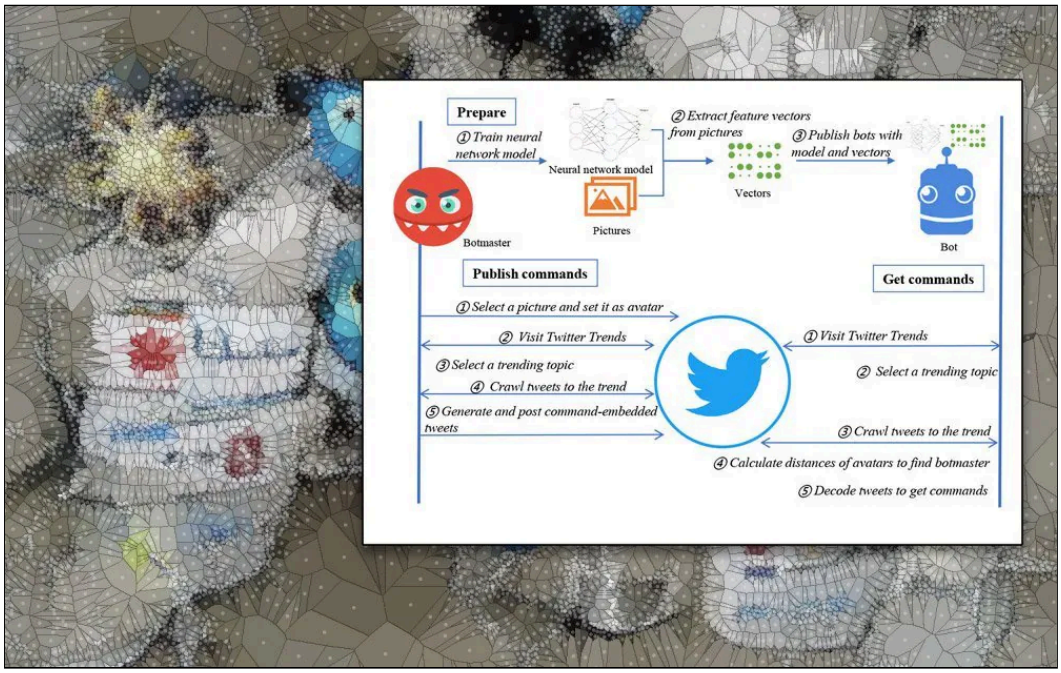
Protecting Twitter-Based Botnet Command and Control Servers With Neural Networks

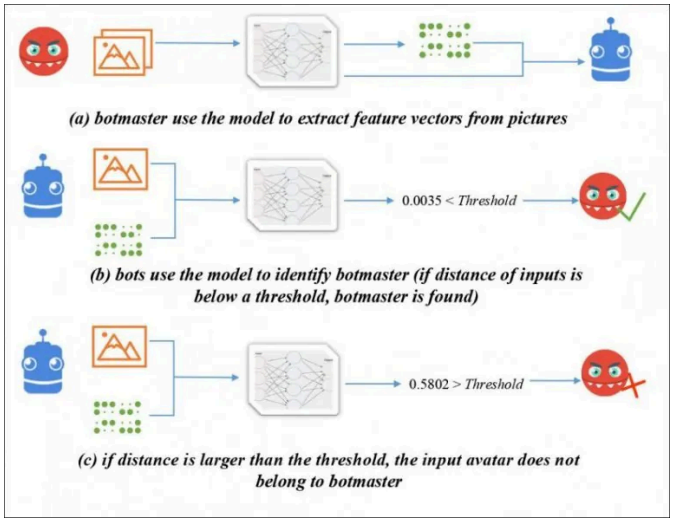
Originally published at https://www.unite.ai/protecting-twitter-based-botnet-command-and-control-servers-with-neural-networks/



Researchers from China have used the 'black box' nature of neural networks to devise a novel method for malicious botnets to communicate with their Command and Control (C&C) servers over Twitter in a way that can't be exposed by security researchers, and could make it difficult to shut down their operations.

The updated paper released on 2nd August is called *DeepC2: AI-powered Covert Botnet Command and Control on OSNs*.

The proposed method, successfully tested in a trial over Twitter, trains a neural network to identify the Twitter account controlled by a C&C entity based on its Twitter user icon. Once the commanding account has been 'authenticated' by the neural network, the hidden commands in its apparently innocuous tweets can be executed by the army of computers that have been infected with the botnet.



Source: https://arxiv.org/abs/2009.07707

A botnet is a group of computers that have been compromised so that they can be marshaled by centrally-located bad actors into performing various types of crowd-sourced cyber-attacks, including DDoS incursions, cryptocurrency mining and spam campaigns.

The Struggle For C&C Anonymity

Each compromised computer in the botnet 'orchestra' requires central direction from the originators of the malware, and thus has to communicate in some way with a C&C server. However, this has traditionally meant that security researchers were able to reverse-engineer the individual botnet infection and expose the URL/s of the C&C servers, usually hard-coded into the infection.

Once the malign C&C domain was known, it was then possible to block it at a network level, and to investigate its origins in pursuit of legal action.

In recent years the trend for C&C servers has veered away from dedicated http domain-based addresses towards the use of popular web services such as Gmail, Twitter, online clipboard services and a variety of online social networks (OSNs).

In 2015 it was revealed that the malware backdoor Hammertoss was using Twitter, GitHub and cloud storage services for these purposes; in 2018, that the Remote Administration Tool (RAT) HeroRat used the Telegram messaging protocol to the same ends; and that in 2020 the Turla Group's ComRAT malware had migrated to using Gmail as a communications framework.

However, these approaches still require some kind of identifying information to be hard-coded into the infecting software, making it discoverable, albeit often with some difficulty, by security initiatives. In such cases the explicable nature of the malicious commands and the identification of user IDs can allow these channels to be shut down, usually disabling the dependent botnet.

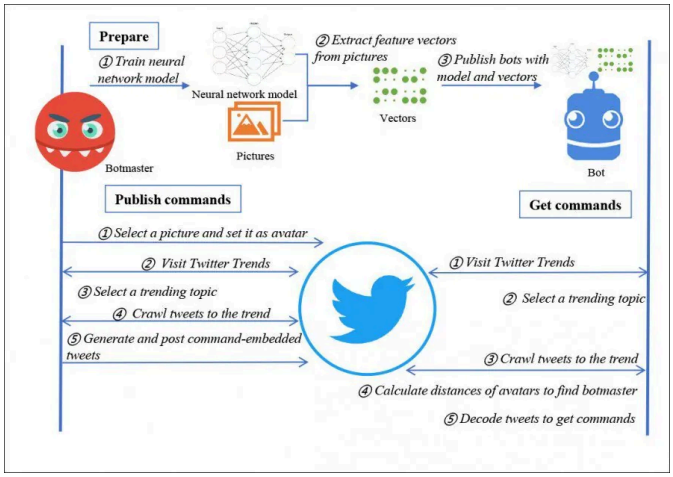
Secret Identifiers

The DeepC2 method proposed by the Chinese researchers makes reverse engineering of the C&C identifying information practically impossible, since all that the code will reveal is an opaque neural network algorithm that can't easily be re-implemented in its highly optimized (i.e. effectively 'compiled') form.

Under DeepC2, the bot locates the 'botmaster' by seeking out a specific user avatar (in trending topics, so that the bot does not need to trawl all of Twitter) whose high-level features have been encoded into its neural network. The authors of the malware select suitable icon pictures in advance of the campaign deployment and train the

neural network on these. The derived feature vectors and the neural network itself are distributed as part of the malicious payload.

When the C&C server publishes a new command, the guiding hand behind it chooses some trending topics on Twitter (or whichever social network is being exploited) and generates apparently ordinary social media posts which contain embedded commands. The topicality of the trend pushes the posts to a discoverable prominence, so that the bots can easily access new commands by crawling the social network.



Workflow of DeepC2.

To prevent detection, each Twitter icon and associated account is used as a C&C mechanism only once under DeepC2, with the system moving on to a fresh pre-rolled C&C OSN handle after this event. Additionally, the bots will delete the used vector information that helped them identify C&C avatars after one use (one command sent), to further impede replay by forensic security methods.

Obfuscating C&C Commands

As a further aid to obfuscation, DeepC2 includes a method to prevent the detection of explicit commands in Twitter messages, by using hash collision and enhanced data augmentation (EDA), the latter based on work out of Protago Labs Research in 2019, in collaboration with Dartmouth College and Georgetown University.

A hash collision occurs when two different pieces of data have an identical checksum, i.e. when each distinct piece of data is mathematically equivalent in profile – an infrequent scenario that can be exploited, in this case, to create command signifiers from apparently innocent text content.

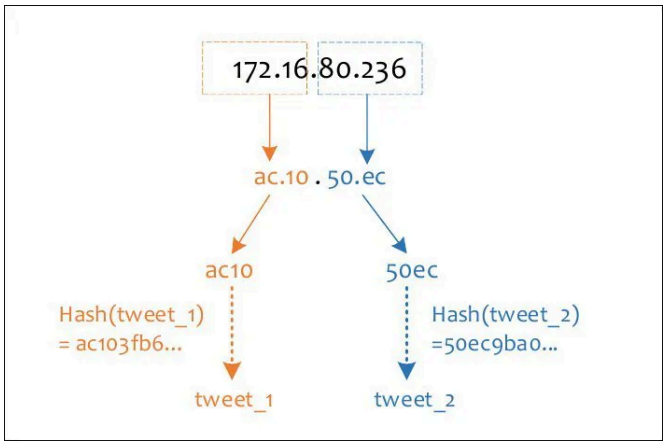
The bots search for these pre-programmed hashes in the social media output of accounts that it has been able to identify as C&C servers based on the recognized avatar icons. Since the tweets generated by the C&C commander will have some contextual relevancy to the target topic stream, they are difficult to identify as anomalies, concealing the intent of the posts.

Operation	Sentence
None	Our TAXII server is going to be taking a short nap at 11am ET today for an update.
SR	Our TAXII server is ***endure*** to be taking a short nap at 11am ET today for an update. Our TAXII server is going to be ***conduct*** a short nap at 11am ET today for an update.
RI	Our TAXII server is going to be taking a short nap at 11am ***cat sleep*** ET today for an update. Our TAXII server is going to be taking a short ***circuit*** nap at 11am ET today for an update.
RS	Our ***short*** server is going to be taking a ***TAXII*** nap at 11am ET today for an update. Our TAXII server is going to be ***today*** a short nap at 11am ET ***taking*** for an update.
RD	Our server is to be taking a short nap at 11am ET today for an update. Our TAXII server is going to taking short 11am ET today for an update.

SR: synonym replacement. RI: random insertion.
RS: random swap. RD: random deletion.

Though the augmented text data may not be grammatically correct, the grammatical inconsistency of posts on Twitter (and other social media networks) effectively hides these 'glitches' in comprehensibility.

IP addresses are conveyed by the botmaster to the bots by splitting the URL into two separate hashes with hash collision, which are identified and concatenated into a comprehensible IP address by the remote bots.



The researchers used seven virtual private servers to simulate geo-diverse locations. Target avatars were derived from 40 photos taken by mobile phones, which were then converted to vectors during training. The bots were subsequently put into position with the trained model and the vector data.

All commands in the experiment were successfully received and parsed by the virtualized bot networks, though with some redundancy of message dissemination, since the system cannot be entirely certain that each instance of a message will be received from any particular tweet.

In terms of countermeasures, the researchers note that the automated frequency of the way that 'slave' bots will trawl Twitter for C&C messages, and the way that the C&C server will iterate through a series of posts, could potentially represent an identifiable signature that could be addressed by new types of protective frameworks.

Furthermore, OSNs could potentially calculate the very specific visual differences which are baked into an array of C&C avatar icons, and develop methods to raise alerts based on those criteria.

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai