POSTER: BotFlex: A Community-driven Tool for Botnet Detection

Sheharbano Khattak
Independent Researcher
sheharbano.k@gmail.com

Zaafar Ahmed,
Affan Syed
SysNet, National Univ. of Computer & Emerging Sciences, Pakistan
firstname.lastname@sysnet.org.pk

Syed Ali Khayam
PLUMgrid, Inc.
akhayam@plumgrid.com

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

Keywords

botnet, network security, correlation

1. INTRODUCTION

Existing botnet detection tools suffer from two primary limitations: case specificity and rigidity. Case-specific detection mechanisms leverage instance-specific characteristics of botnets (typically related to C&C). As a result, the scope of detection is narrow and tailored to a sub-class of the entire phenomenon. Similarly, rigidity in design leads to difficulties in (a) accommodating new detection parameters in their decision policies, (b) tuning/configuring their operation, (c) modification of decision policies, and (d) simultaneous implementation of multiple decision policies. In addition to these architectural aspects, flexibility of the tools is further compromised by unavailability of source-code. Currently, no tool exists that can be used to easily implement and test new botnet detection mechanisms.

In this poster we present BotFlex— a community-driven network-based tool for botnet detection, designed to address the shortcomings of existing tools identified above. We also present our first-cut implementation of BotFlex which conceptualizes botnet infection as a complex event detectable via multiple trigger paths. BotFlex employs a custom correlation framework which detects and curbs botnet threats by reasoning about vertical (across time) and horizontal (across multiple entities) events.

Both BotFlex and its correlation framework are released in open-source for community use [2]. We evaluate BotFlex for accuracy over 500 GB of enterprise traffic collected from one of Pakistan's largest ISPs, Nayatel, with ground truth obtained using a one-time sample of Team Cymru's reputation list for the collection duration. By tuning different parameters, BotFlex demonstrates a better detection accuracy with lower false positives than our baseline evaluation tool BotHunter [3]. Despite encouraging preliminary results, we acknowledge that BotFlex is still in its infancy and hope that its built-in design flexibility and extensibility will allow the community to extend it into an effective and actively updated tool.

CCS'13, November 4–8, 2013, Berlin, Germany.
ACM 978-1-4503-2477-9/13/11.
http://dx.doi.org/10.1145/2508859.2512507

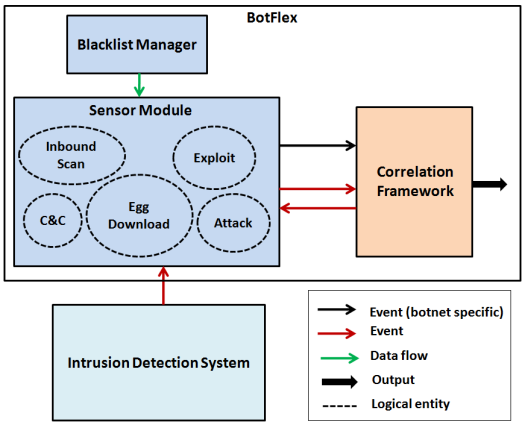

Figure 1: BotFlex architecture.

2. BOTFLEX–ARCHITECTURE AND IMPLEMENTATION

BotFlex is a network-based botnet detection tool, designed to be (i)*domain specific* and *community-driven* with a view to conveniently develop, improve upon, and/or benchmark existing and new botnet detection solutions, (ii)*flexible* in fine tuning its detection thresholds and conditions to cater to varying organizational/deployment accuracy and delay requirements; and *extensible* in easy integration of new detection parameters and decision elements to keep up with the rapidly-evolving botnet threat, (iii)*promptly process information* for early detection of threats and subsequently activate evasive countermeasures, with a (iv)*simple user interface* to define botnet detection policies and thus improve end-user productivity.

BotFlex's architecture (Figure 1) comprises of three modules: the blacklist manager, the sensor module and the correlation framework. We implement BotFlex over Bro [5]. Bro has a layered architecture, where low-level network traffic is incrementally refined to meaningful network events which

can be handled at the top-most scripting layer by Bro's domain-specific scripting language. BotFlex resides at this layer and its sensor module and the correlation framework have been written entirely in the Bro scripting language. We now discuss BotFlex's architectural modules, along with their respective implementation.

Blacklist Manager:. The blacklist manager complements the sensor module in its operation by providing it with up-to-date intelligence, such as C&C and The blacklist manager has been implemented as a bash script that downloads a number of public blacklists [2], organizes the intelligence based on its subject (IP, URL, subnet, port) and normalizes it according to a specific format. BotFlex reads the blacklists into a fast data structure that is synchronized with the back end blacklists. The blacklist manager is flexible in that it can be replaced with any other service as long as the blacklists adhere to the file format used by BotFlex.

Sensor Module:. The sensor module generates symptoms of botnet infection as events derived from the underlying NIDS–Bro, thus allowing for information to be processed and reacted to as it is churned. Events produced by sensor module may be readily consumed (simple events) or derived by further processing (derived events). Derived events can optionally involve one or more iterations through the correlation framework. Based on the well-known [1, 3] bot lifecycle events, the sensor module eventually maps all simple and derived events to five high-level activity classes: inbound scan, host exploit, malicious binary (egg) download, C&C communication and outbound attack. Currently, BotFlex uses a preliminary list of botnet detection parameters from existing literature . The sensor module is extremely flexible as it uses tunable attributes; the threshold values used to trigger various events, their weights and observation intervals (or time windows) are configurable. Furthermore, being independent of the correlation framework, the sensor module supports addition/removal/modification of detection parameters as threats evolve without requiring modifications to the correlation logic.

Correlation Framework:. The correlation framework continuously receives events from the sensor module and correlates them according to rule(s) specified by the user to derive the complex event of botnet infection. Note that we input botnet-related events to the framework, but in principle it is an independent, self-contained entity capable of processing any events fed to it.

The correlation framework has been implemented in Bro scripting language as a Complex Event Processing (CEP) engine. The present problem of botnet detection clearly conforms to the CEP model where events are defined with respect to time, causality and aggregation [4]. CEP allows correlation of events in real-time to detect a target complex event comprising of multiple simple or complex events. The choice of CEP makes BotFlex (i) flexible in how sensor module alerts are correlated; and (ii) extensible in facilitating addition of new correlation conditions; (iii) with a faster response because of a temporally-aware, event-driven model where information is processed as soon as it is churned and discarded when it is no longer relevant/needed. We have implemented a declarative correlation *rule language* that allows flexible processing of events related to botnet infection.

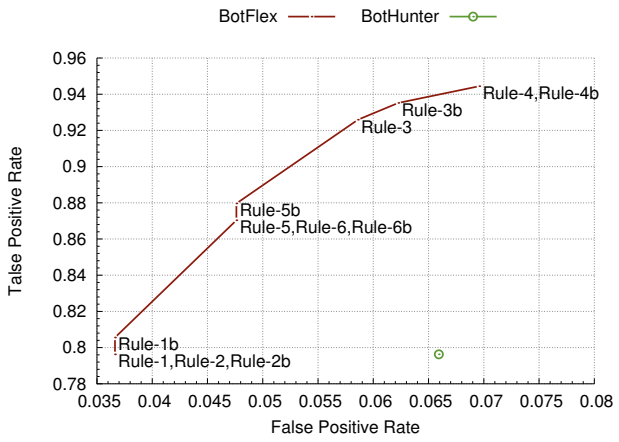

Figure 2: ROC curve for botnet detection with BotFlex and BotHunter

Correlation framework can handle both vertical and horizontal correlation. [1] The correlation framework is novel in that it marks the first time a CEP engine has been built within a non-proprietary NIDS with a view to accelerate information processing (CEP engine is typically treated as an external entity to which NIDS alerts are fed). Our approach has two main advantages: (i) CEP engine built within NIDS can directly ingest NIDS events and data structures thus reducing the delay incurred due to translation of NIDS events to an intermediate format when working with an external CEP engine, (ii) NIDS can readily understand and correlate events derived from disjoint sources.

3. PRELIMINARY EVALUATION

3.1 Evaluation Dataset and Methodology

We evaluate BotFlex for accuracy over a 500 GB data trace obtained from one of Pakistan's leading ISPs, Nayatel. The traffic contains 48,606 unique IP addresses of which 381 IP addresses represent Nayatel's local network statically assigned to mid-to-small size enterprises. While we assume that an IP address represents a single machine, we acknowledge that the ISP customers are likely using their public IPs to represent multiple private (NATed) hosts. However, this assumption of a one-to-one mapping between public IPs and bots provides a realistic test environment for deployment of a real-world bot detection solution.

We obtain ground truth for this data from a one-time sample of the IP reputation feed provided by Team Cymru. The ground truth comprises a proprietary threat repository of known C&C servers identified with the help of a chain of globally-deployed sensors. These C&C servers (referred to as the ground truth blacklist henceforth) were being actively contacted by hosts from the monitored region during our data collection at the ISP's B-RAS (between 08:54:49 AM and 04:13:29 PM PKT on September 18th, 2012). We labeled our data on the principle that the hosts in our dataset that communicate with the ground truth C&C servers are bots. The ground truth identified 108 (28.3%) of the total

[1]Vertical correlation performs temporal analysis of events generated for a single entity. Horizontal correlation deals with spatial analysis of an event pattern for multiple entities.

Table 1: Description of BotFlex correlation rules.

Name[a]	Rule	Description
1	(cnc_blacklist OR cnc_other) AND (exploit OR egg OR attack)	C&C and any other malicious activity
2	cnc_blacklist OR (cnc_other AND (exploit OR egg OR attack)	Same as 1, with a direct trigger for C&C blacklist match
3	(exploit OR egg) AND (cnc_blacklist OR cnc_other OR attack))	Evidence of any inbound host compromise and outbound C&C or attack
4	cnc_blacklist OR ((exploit OR egg) AND (cnc_other OR attack)))	Same as 3, with a direct trigger for C&C blacklist match
5	(egg AND (cnc_blacklist OR cnc_other)) OR (attack AND (cnc_blacklist OR cnc_other)) OR (egg AND ATTACK)	Egg download and outbound C&C or attack, or C&C and attack
6	cnc_blacklist OR (egg AND cnc_other) OR (attack AND cnc_other) OR (egg AND attack)	Same as 5, with a direct trigger for C&C blacklist match

[a]Rule{1,...,6}b variations include an OR rule for hosts tagged as part of a spam or scan campaign through horizontal correlation

381 IP addresses in the data trace as compromised bots.

3.2 Accuracy evaluation

We take a bottom-up approach in evaluating BotFlex in terms of accuracy. Thus, we first identify the best thresholds for the sensor module, and then compare different correlation rules to identify the most effective bot detection system. We run BotHunter [3] on the same dataset as a baseline to validate our results. For fair evaluation, we do not provide the ground truth blacklist to either of BotHunter or BotFlex. We first observe the impact of different threshold values for threshold-based detection parameters in the sensor module on bot detection. We then use best ROC operating points to identify suitable thresholds. We acknowledge that the use of a bot-labeled ground truth to measure accuracy of specific sensors (spam, scan etc.) is not perfect; however it remains a more rigorous approach than setting arbitrary thresholds.

After establishing suitable settings for the sensor module, we formulate correlation rule(s) (Table 1) for botnet detection to identify the strengths and weaknesses of various correlation policies. We plot the result of each run on an ROC curve. Also, note that we removed the detection parameter 'inbound scan' from our rules after noticing that it is nearly omnipresent (triggered for 341 of our total 381 ISP local hosts) possibly because of legitimate applications (e.g. p2p peer discovery) and Internet background noise.

We now discuss the insights gathered from the ROC curve for correlation rules (Figure 2) . Vertical correlation rules indicate that relying heavily on evidence of C&C communication produces low detection rates. This can be improved by complementing C&C evidence with other detection parameters. For rules Rule{1,2...,6}b involving horizontal correlation, we detect 23 hosts through coordination in spam-like activties and another 17 are detected based on synchronization in outbound scan timings. We generally find that horizontal correlation increases TP with no effect on FP with the exception of one host in Rule 3b (the latter was found to be a spambot later through manual investigation). Hence, horizontal correlation can possibly identify previously undetected bots based on their activity coordination with other botnet members.

To sum up, BotFlex (with Rule 3) detects 100 of the ground truth 108 bots, with a TPR of 92.6% and an FPR of 5.8%. These results are comparable with our baseline tool, BotHunter, which on the same trace gave a TPR of 79.6% and an FPR of 6.6%. Note that FPR for both the tools could represent an inflated number as our ground truth is biased in favor of TP (sensitivity) at the expense of TN (specificity) (the source of our ground truth claims to have nearly zero false-positives from their IP reputation feed).

4. CONCLUSION/FUTURE DIRECTIONS

The botnet research community currently lacks an open-source and community-driven tool to develop with ease, improve upon, and/or benchmark existing and new botnet detection solutions. In this paper, we presented BotFlex–a domain-specific, flexible and extensible network-based tool for botnet detection. We evaluated BotFlex for accuracy while comparing with a relevant baseline tool, and found the results to be encouraging. Our next research goal is to evaluate BotFlex's performance and scale it to increased traffic volumes while also taking its underlying NIDS platform into account. Additionally, BotFlex will also benefit from extension of its sensor module alerts and correlation rules, and enhancement of the correlation framework's custom rule language for defining correlation policies.

5. REFERENCES

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 41–52, New York, NY, USA, 2006. ACM.

[2] BotFlex. `http://sysnet.org.pk/BotFlex`. Online. Feb,2013.

[3] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *Usenix Security Symposium*, 2007.

[4] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[5] V. Paxson. Bro: a system for detecting network intruders in real-time. In *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, SSYM'98, pages 3–3, Berkeley, CA, USA, 1998. USENIX Association.