Distributed Crawler Detection in Peer-to-Peer Botnets

(Technical Report IR-CS-77)

Dennis Andriesse
VU University Amsterdam
The Netherlands
d.a.andriesse@vu.nl

Christian Rossow
Saarland University, Germany
crossow@mmci.uni-saarland.de

Herbert Bos
VU University Amsterdam
The Netherlands
h.j.bos@vu.nl

ABSTRACT

This addendum provides additional details on the distributed crawler detection algorithm described in our full paper on recon in Peer-to-Peer (P2P) botnets [1]. Specifically, we provide a discussion of the algorithm's tolerance to adversarial nodes and Sybil attacks.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Invasive Software*; C.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*

General Terms

Security

Keywords

Peer-to-Peer Botnet, Crawling, Reconnaissance

1. BYZANTINE FAULT TOLERANCE

This addendum details the Byzantine-robustness of the distributed crawler detection algorithm described in our full paper on the subject [1]. We evaluate the algorithm's tolerance to injection of Byzantine nodes (or Sybils) which attempt to skew the algorithm outcome in the host botnet. Specifically, we consider the following Sybil attacks. (1) Sybils may be injected in unison with crawling efforts to prevent crawlers from being detected. (2) Conversely, Sybils may be used to coerce the crawler detection algorithm into blacklisting legitimate bots. This section describes the algorithm's tolerance of these attacks. Note that centralized versions of our algorithm are not vulnerable to Sybil attacks [1], though they do require a hybrid centralized/distributed architecture of the host P2P botnet.

To obtain full control over the algorithm's decisions, an attacker must control the majority of group leaders. To prevent such attacks, we designed our algorithm such that the botmaster randomly selects new group leaders for each round [1]. Thus, attackers cannot predict which nodes to take over or otherwise attack in order to dominate the group leaders. Nevertheless, if an adversary injects enough Sybils into the network, the probability that some of these are selected as leaders (as part of the normal selection process) becomes non-negligible. While the leader majority voting process prevents an individual compromised leader from doing harm, an excessive Sybil population may succeed by taking over a majority of leaders. An adversary has full control over the algorithm outcome if he controls $|M| > |G| \times v$ out of $|G|$ leaders, where M is the set of all Sybil nodes in the botnet, G is the set of groups, and v is the voting threshold (for a minimal majority vote, $v = 50\%$).

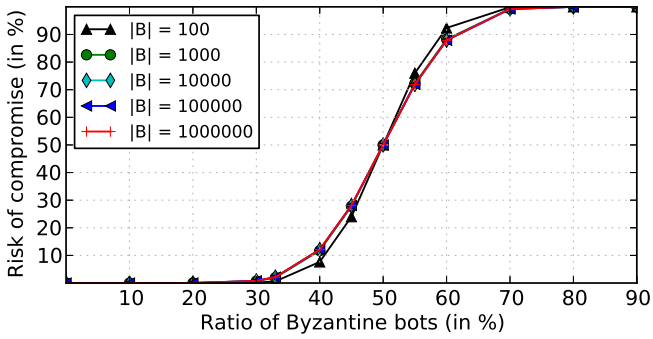
1.1 Probabilistic Risk Model

We use a statistical evaluation to compute the risk that a group leader population is dominated by malicious nodes. This section describes the probabilistic risk model we use, while Section 1.2 applies our model to derive the risk of compromise in a full-scale application of our crawler detection algorithm.

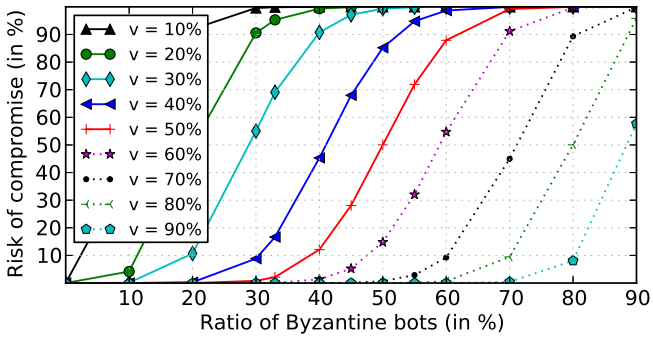
The probability that a malicious peer is (randomly) selected as a group leader is $|M|/|B|$, where B is the set of all bots (including the malicious ones) in the network. With $|G|$ groups, there are $\binom{|B|}{|G|}$ possible combinations of group leaders. To compute the risk of compromise, we must therefore find all combinations that contain more than $v\%$ malicious bots.

For instance, given $|G| = 3$, $v = 50\%$, $|B| = 10$ and $|M| = 4$, there are $\binom{10}{3} = 120$ possible group leader combinations. To dominate the leader population, at least two group leaders must be malicious. We call such combinations *compromised*. We separately compute the probability of compromise for every majority of n malicious leaders. For $n = 3$, there are $\binom{6}{0} \times \binom{4}{3} = 4$ compromised combinations, while for $n = 2$, there are $\binom{6}{1} \times \binom{4}{2} = 36$ compromised combinations. Thus, for the given parameters there are a total of $4 + 36 = 40$ compromised combinations out of 120 overall, so that the chance that an adversary gains control over a voting round is $1/3$.

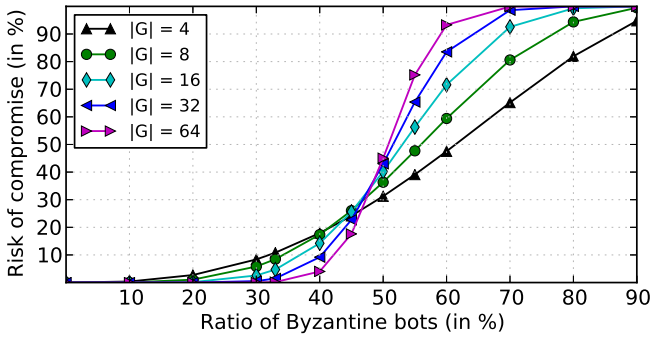
This example can be generalized as follows. There are $\binom{(|B|-|M|)}{(|G|-n)} \times \binom{|M|}{n}$ combinations for which exactly n group leaders are malicious. The overall number of



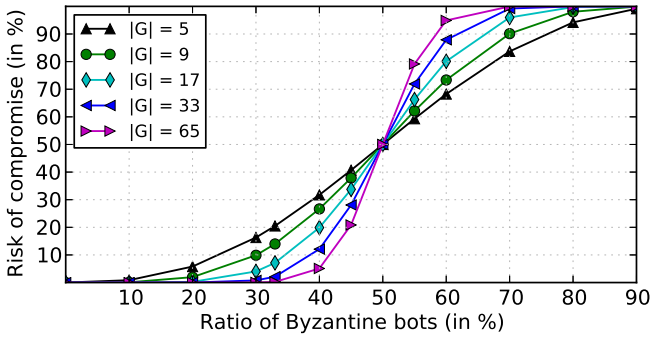
(a) Varying botnet sizes.



(b) Varying voting thresholds.



(c) Varying number of groups (even numbers).



(d) Varying number of groups (odd numbers).

Figure 1: Risk of compromise (y-axis) versus ratio of attacker-controlled bots (x-axis) for different algorithm settings.

compromised combinations is the sum of compromised combinations for each $n \in [|G| \times v + 1, |G|]$. The probability $P(c)$ of randomly selecting a compromised combination is then the number of compromised combinations divided by the number of overall combinations, as shown in Equation 1.

$$P(c) = \frac{1}{\binom{|B|}{|G|}} \sum_{n=(|G|\times v)+1}^{|G|} \binom{(|B| - |M|)}{(|G| - n)} \times \binom{|M|}{n} \quad (1)$$

1.2 Probabilistic Risk Evaluation

This section computes the risk of compromise in a full-scale deployment of our crawler detection algorithm for various configurations. Figure 1a shows the probability of compromise versus fraction of malicious bots ($|M|/|B|$) for varying botnet size. We fix $v = 50\%$ and $|G| = 33$ and vary the botnet size $|B|$ from 100 to 1,000,000. As can be seen from the graph, botnet size does not significantly impact the probability of compromise. Regardless of botnet size, the probability remains 0.7% if attackers control 30% of the bots.

Figure 1b shows the chance of compromise for $|G| = 33$ and $|B| = 100,000$ (a typical P2P botnet size) with varying voting threshold v. Clearly, lower v means that an adversary needs to control fewer malicious bots to manipulate the algorithm outcome, while higher v increases the resilience against intruders. For instance, a defensive implementation of our algorithm could choose $v = 90\%$, so that even if half of the nodes are Sybils, the risk of compromise is just 0.00007%. In general, choosing v remains a trade-off between accuracy and completeness (including detection of low-coverage crawlers), and robustness against intruders.

Figure 1c shows how the number of voting groups ($|G|$) influences the robustness of the algorithm for $|B| = 100,000$ and $v = 50\%$. The number of groups in Figure 1c is even, which means that exactly half of the votes ($|G|/2$) does not constitute a majority. In the special case of $v = 50\%$, this changes for odd numbers of voting groups, as shown in Figure 1d. This causes the risk of compromise to be slightly higher, as relatively fewer nodes are required to form a majority. It can be seen that smaller groups (i.e., larger values for $|G|$) are favorable, as this minimizes the risk of compromise if an attacker controls slightly under $v \times |B|$ bots. For instance, for an attacker that controls 45% of the bots, the risk of compromise is 40.7% for $|G| = 5$, but only 20.9% for $|G| = 65$.

1.3 Practical Risk Assessment

Section 1.2 has shown that despite our use of majority votes, an adversary can still probabilistically compromise the algorithm with a minority of bots, although their chance of success in any particular detection round may be low. To see how this can escalate over multiple detection rounds, consider a practical instance of our algorithm in the GameOver Zeus botnet (taken from our full paper on P2P botnet recon [1]), where $|G| = 8$, $v = 50\%$ and $|B| \approx 200,000$. An attacker who injects 50,000 Sybils controls 25% of the nodes and can compromise the algorithm with a probability of about 1%. While an individual detection round is unlikely to be compromised, the probabilistic attack is likely to eventually succeed in at least some rounds (given enough time and without further hardening). In this particular example, given hourly detection rounds, there is an 82% chance that an adversary can compromise at least one detection round in a time span of one week. Even without resorting to a centralized implementation of our algorithm, several approaches exist to reduce the risk or effect of such compromise.

Prior work has proposed strategies to limit the damage potential of Sybil attacks, typically by establishing a chain of trust or reputation scheme to complicate the insertion of adversarial nodes [5, 2]. Such techniques can be combined with our algorithm, though at the expense of simplicity of deployment.

Alternatively, heuristic approaches can also reduce the risk of Sybil attacks to a limited degree. For instance, botmasters can complicate Sybil attacks by limiting the number of leaders selected from the same network block. Similarly, leaders can be selected using a reputation mechanism like that used in Sality [3], or according to a proof-of-work scheme as proposed by Hund et al. [4]. Note that it is crucial that sufficient randomness is preserved in leader selection to prevent targeted attacks against future leaders.

While our algorithm is capable of fully automated crawler detection, the above suggests that it may not be wise to automatically blacklist crawlers detected in a particular round — doing so would allow a single compromised round to escalate into a blacklisting attack against the botnet itself. Instead, the decision can be based on multiple successive detection rounds, or even on manual evaluation of the results by the botmasters. Also note that for an adversary to permanently hide a crawler, they must compromise most or all of the detection rounds, requiring a far greater number of Sybils than is needed to compromise a single round.

Acknowledgements

This work was supported by the European Research Council through project ERC-2010-StG 259108 “Rosetta”.

2. REFERENCES

[1] D. Andriesse, C. Rossow, and H. Bos. Reliable Recon in Adversarial Peer-to-Peer Botnets. In *IMC'15*, 2015.

[2] J. Dinger and H. Hartenstein. Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self-Registration. In *ARES'06*, 2006.

[3] N. Falliere. Sality: Story of a Peer-to-Peer Viral Network, 2011. Tech report, Symantec.

[4] R. Hund, M. Hamann, and T. Holz. Towards Next-Generation Botnets. In *EC2ND'08*, 2008.

[5] H. Rowaihy, W. Enck, P. McDaniel, and T. la Porta. Limiting Sybil Attacks in Structured P2P Networks. In *INFOCOM'07*, 2007.