Pervasive Random Beacon in the Internet for Covert Coordination

Hui Huang Lee, Ee-Chien Chang and Mun Choon Chan

School of Computing
National University of Singapore
{leehuihu,changec,chanmc}@comp.nus.edu.sg

Abstract. A random beacon periodically outputs a random number and was introduced by Rabin[12] to secure remote transaction. We consider a random beacon that is *pervasive* in the sense that, it is available everywhere, and accesses to the beacon blends with normal activities. With a pervasive beacon, it is difficult to disrupt the beacon and detect accesses to it. As a result, the pervasiveness of the beacon can facilitate covert coordination, whereby a large collection of agents covertly decide on a common action. In this paper, we discuss the desirable properties of a pervasive random beacon which can be used for covert coordination, and describe how such a beacon can be found in the Internet based on major stock market indices closing values. We also investigate how such a covert coordination can be used, in particular, in coordinating distributed denial of service (DDoS) attacks. Finally, we explore ways to, in a limited manner, disrupt the beacon.

1 Introduction

A random beacon periodically outputs random bits and was introduced by Rabin [12] to secure remote transactions such as contract signing. Since then, a number of other applications of random beacons have been proposed. For example, Bennett et. al. proposed using a random beacon to authenticate video recording[2]. Aummann and Rabin [1] also proposed using higher bandwidth beacon to achieve unconditional security with respect to eavesdroppers with limited storage. Mossel and O'Donnell investigated methods of obtaining a random beacon from a noisy source[10]. Additional discussions on random beacons can be found in [3, 6].

In this paper, we introduce an additional requirement of *pervasiveness*, and give a construction of pervasive random beacon by using information available in Internet, namely major stock market indices closing values. The advantages of using this random source is that it is widely available and replicated on many web servers. Furthermore, there are enormous accesses of this information from vastly different URLs. As a result, it is difficult to distinguish accesses to the beacon from normal web activities. Disrupting access to the beacon is also difficult without substantial disruption to normal web accesses.

As an application of such a pervasive random beacon, we demonstrate how it can be used to coordinate DDoS attacks. In a distributed denial of service attack (DDoS), an attacker employs multiple machines (also known as agents or zombies) to attack a victim, preventing it from providing services to legitimate clients.

Existing coordination models in DDoS attacks can be grouped into three categories: *manual*, *semi-automatic* or *automatic* [8]. In a manual or semi-automatic attack, the attacker (or master) send the attack parameters such as the network address of the victim and the time of attack to the agents (or zombies). The attacker can directly send his commands to the agents, or the communication can be indirect through another layer of proxies. One weakness of the manual and semi-automatic attacks is that the discovery of one entity may lead to the discovery of the DDoS network.

Alternatively, a DDOS attack can be automated by avoiding communication among the agents and attacker altogether, and thus reduces the risk of detection. However, the parameters of an attack, including start time, attack type, target, are preprogrammed in the code. As a result, once a copy of the agents is captured and examined, parameters of the attack will be known, and usually well in advance of the attack.

In view of the above, we look into whether other models of covert coordination can be employed by the attackers, such that the discovery of an agent will not reveal the attack parameters, and hence will not compromise the DDoS network. It turns out that this can be easily achieved if a pervasive random beacon is available.

In the rest of this paper, we will investigate a method that uses stock closing indices to provide a pervasive random beacon. In Section 2, we discuss various desirable properties of a pervasive random beacon. In Section 3, we describe implementation issues in using the stock closing indices. A survey on current DDoS coordination models is given in Section 4. An alternative model is proposed in Section 5. In Section 6, we describe a few potential methods to disrupt such a beacon.

2 Pervasive Random Beacons

A random beacon periodically outputs random bits. There are a few formal formulations of randomness. In this paper, we take an informal description: the outputs cannot be computationally distinguishable from an uniform distribution. In addition, the output has to remain unpredictable, until the time the random number is revealed. Hence, a secure pseudo random number generator by itself is not sufficient to be a random beacon. An additional infrastructure is required to ensure that the random numbers are honestly and periodically generated, for example, a trusted provider that periodically outputs a random bit using a secure pseudo random number generator, can be a random beacon.

In this paper, we consider random beacons that are pervasive. There are two additional requirements:

High Availability: We require that the outputs of the random beacon can be easily obtained most of the time. Hence, a trusted provider that outputs random bits may not be pervasive if it is the only source. On the other hand, if the outputs from the provider are extensively replicated and the copies are publicly available, the beacon can be pervasive.

Blended Access: We require that accesses to the beacon can be blended with normal activities, making it difficult to distinguish beacon accesses from normal activities.

When a beacon is available in many locations to provide high availability, accesses to the beacon can also be distributed over a large number of servers. Together with the ability to blend with normal activities, it is very difficult to identify beacon accesses or disrupt the beacon. These properties facilitate covert operations.

3 WWW content as Pervasive Random Beacon

To find a pervasive random beacon in the Internet, we look in the WWW and consider content-based random sources, for example lottery results, political events and sport events. After exploring various possibilities, we found that the stock closing indices are good candidates for the beacon. First, they are replicated all over the WWW and widely accessed. Furthermore, it is well-accepted that a stock index can be used as a random source, for example, there are also other works that use stock index as random seed[5].

A stock market index is calculated using a certain number of stocks from its stock market. For example, the Dow Jones Industrial Average (DJIA) is a price-weighted average of 30 blue-chip stocks that are typically traded on the New York Stock Exchange. During trading period, the value of an index fluctuates, and the reported value can be inconsistent among different service providers at any time. On the other hand, the daily closing index is static and consistent throughout the market's closing period. Since different stock markets around the world have different closing times, by using several indices from different stock markets, we can obtain several random bits, each at a different time of the day.

After deciding on using the stock indices, there are two implementation issues. Firstly, how many random bits can be extracted from a stock index. Secondly, how should the beacons be accessed. We will discuss these issues in the rest of this section.

3.1 Stock indices as random beacon

Since different stock markets around the world have different closing times, by using several indices from different stock markets, we can obtain several random bits, each at a different time of the day. As an illustration, we can use the following 4 indices (All times stated will be in Coordinated Universal Time (UTC)):

1. Dow Jones Industrial Average (DJIA): The DJIA comprises of 30 components and is from the New York Stock Exchange (closing period is from 21:00 to 14:30 when daylight saving time is not in effect and from 20:00 to 13:30 if daylight saving time is in effect)
2. Nikkei 225 (N225): The N225 comprises of 225 components and is from the Tokyo Stock Exchange (closing period is from 06:00 to 00:00)
3. Straits Times Index (STI): The STI comprises of 45 components and is from the Singapore Stock Exchange (closing period is from 09:00 to 01:00)
4. FTSE 100: The FTSE 100 comprises of 102 components and is from the London Stock Exchange (closing period is from 16:30 to 08:30 when daylight saving time is not in effect and from 15:30 to 07:30 if daylight saving time is in effect)

Figure 1 show the closing period of the 4 stock exchanges. Closing stock quotes for major indices are stored and available on the web, e.g. DJIA is available starting Oct 1, 1928 from `quote.yahoo.com`. Hence, it is not necessary to get them during the closing period.

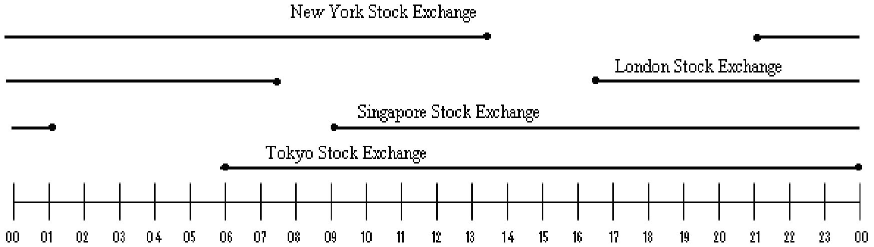

Fig. 1. Closing hours of the 4 stock exchanges

From an index, a mixing function is applied to extract a few bits. An example of a mixing function is a series of XOR operations on the binary representation. Ideally, the number of bits extracted should be the entropy of the closing index. In order to obtain an estimate of the entropy value, we use a publicly available random tester `ent` [14], which provides an estimate of the entropy of a given set of input data.

The test is performed on DJIA closing index (round to 2 decimal places) for the past 30 years. Only the 15 least significant bits are used in the test. The random tester `ent` determines that the entropy is about 13 bits. More random bits can be obtained by considering individual stock or other indices in the same market.

3.2 Accessing the beacon

Major stock market closing indices can be found on many online newspapers as well as websites of financial organizations. In an implementation, a list of websites can be preprogrammed. For each access, a website is chosen and its web-pages is parsed to obtain the necessary information. To ensure consistency, two or more websites can be visited and parsed. A more sophisticated access mechanism may use web indexing services or web-search engine to update the preprogrammed list.

In order to evade detection, access to the beacon has to be camouflaged and blended into normal network activities. One way to achieve that is to mimic normal web-surfing behavior. In our implementation, randomness is introduced into the access pattern. For instance, each agent will prefer a particular website, but it will also access the information on some other randomly chosen websites with certain probability.

4 Coordination Model in DDoS

There are several methods an attacker can use to coordinate a DDoS attack. Based on their communication models, current methods can be grouped into three categories: *manual*, *semi-automatic* or *automatic* [8].

In a manual or semi-automatic attack, the attackers send the attack parameters such as the victim and attack time to the agents. The attacker can directly send his commands to the agents, or the communication can be indirect. In the handler-agent model, the attacker sends his commands to a few handler machines which will then relay them to the agents. For example, DDoS tools such as Stacheldraht and Trin00 employ such indirect communications.

One weakness of the manual and semi-automatic attacks is that the discovery of one entity may lead to the discovery of the DDoS network. If a copy of an IRC based tool is captured and examined, the name and password of the IRC channel used by the attacker and agents for communication can be revealed. The network traffic of the captured agents can also be monitored to expose the identity of other agents or handlers. In addition, communication may generates suspicious traffic. For instance, the packets used could be of a specific protocol (E.g. TCP or UDP) and specific port numbers, and the payload of the packets will likely contain strings conforming to a specific syntax. Detection of such suspicious communications among the agents can also lead to the discovery of the network.

An automatic attack avoids communication among the agents and attacker altogether, and thus reducing the risk of detection. However, the parameters of the attack, including start time, attack type, target, are preprogrammed in the code. Examples of such predetermined attacks include the Blaster worm and the Code Red worm (see [4, 11]). Once a copy of the agents is captured and examined, parameters of the attacks will be known. In practice, such discovery usually happens well in advance of the attack. Hence, appropriate countermeasures such as employing extra physical resources (e.g. extra servers or high bandwidth links), or deploying experts at the victim site, can be carried out to mitigate the effects of the attack. For instance, it was discovered on 11th August 2003 that the Blaster worm had been preprogrammed to attack Microsoft's Windows Update website starting from 16th August 2003, thus giving Microsoft ample time to react[7].

Note that although the attacks are preprogrammed, usually a backdoor is still left open for further modification of the code. Nevertheless, communication will be much lower compared to the manual and semi-automatic method.

5 Covert Coordination in DDoS

A DDoS attack based on *covert coordination* would be much harder to defend against. In a covert coordination, a large collection of agents decide when to carry out a synchronized action, and the action to be performed. The coordination is covert in the following ways. (1) Activities, in particular communication among the agents, should be hidden and difficult to distinguish from normal activities. (2) The capture of one agent will not expose the identity of the other agents. (3) Finally, if an agent is captured, the action to be performed, and the time to carry out the action, will not be revealed.

A covert coordination can be achieved by using a pervasive random beacon. The coordination of DDoS can be carried out in the following way. Periodically, the agents obtain two random number r_1 and r_2. Within a weekday, except holiday, 4 random numbers corresponding to 4 major stock markets can be obtained. It is not necessary for the agents to access the beacon at the same time, since archive of the stock indices are readily available. From r_1 and possibly other parameters like the date, the agents determine whether to commence an attack. If an attack is to be launched, using r_2 and a predefined table, the actual time of the attack t, the attack type and the victim are determined.

The total number of random bits per beacon access depends on the design of r_1 and r_2. There is no need for a lot of bits. About 13 bits is sufficient, with 9 allocated to r_1 and 4 allocated to r_2. From the test described in Section 3, the DJIA alone provides 13 bits. If more bits are required, the other market indices can be used as well.

Independent agents: Since each agent does not communicate with other agents or handlers, even if some of them are discovered, no information (e.g. IP addresses) that lead to the discovery of other agents will be revealed. Taking a few discovered agents offline will at most reduce the number of agents available for attacks, and will in no way disable the DDoS network.

Furthermore, the attacker's job is finished after the agent code is installed on the compromised machines. Thereafter, the attacker and the agents do not communicate with each other and hence is virtually impossible to trace based on the network traffic.

Agents remain hidden: Due to the pervasiveness of the beacon, it is difficult to distinguish an agent's beacon accesses from normal web-activities. During the coordination process, the only incoming and outgoing traffic used by agents are normal, well-formed HTTP requests and HTTP replies.

Detecting such activities could be easier in the agents' end, for instance, by an intrusion detection system in or near an agent. Furthermore, it is also easier to detect the agents by scanning the compromised hosts. Note that typically, DDoS attacks are carried out by agents who live in less secure hosts.

However, at the web-server, or any intermediate gateway in an Internet Service Provider(ISP), distinguishing such activities among legitimate usages would be difficult, even with the collaboration of several ISPs. Since the agents cannot be confidently identified, it is difficult to preempt the attack by blocking their web accesses.

Probabilistic attack parameters: If an agent is captured and its program is analyzed, the actual algorithm that determines the attack parameters (the attack time and type) will be revealed. However, the attack parameters will still remain probabilistic, since the beacon is unpredictable. Even if the beacon is closely monitored, the defenders will still have limited time to react. Such uncertainty places the defenders in a stressful situation. For instance, the additional hardwares and experts have to be on standby and be readily deployable for an extended period.

On the other hand, since the attack parameters are probabilistic, the attackers also do not have direct control over the agents and the actual DDoS attacks may not be successful. However, in the context of DDoS, the defenders generally suffer more than the attackers since the defenders have to be prepared for the worst case scenario. For example, the attacker may assign a small probability, say 2^{-11} of commencing attack in the earlier phase. The probability is small so as to provide sufficient time for the DDoS network to grow. Nevertheless, there is still a small possibility that an attack commences early. When the attack is launched too quickly, the chances that sufficient agents have been recruited might be low. When the existence of the agents and risks are known, even though the likelihood and damage may be low, the victims will still have to react immediately to prepare for the small chance that a successful DDoS attack could be launched.

6 Disrupting and influencing the Beacons

In this section, we look at some mechanisms that disrupt the beacons.

6.1 Targeting the reporting services

It is well-accepted that it is difficult to manipulate or predict the stock indices. Furthermore, recall that a mixing function is applied to each index to obtain the random bits. This makes manipulation or prediction even more difficult, since a small perturbation of an index would lead to a different output.

While it is difficult to influence stock indices, it is relatively easier to influence the reporting of the indices. For instance, with sufficient incentive, a financial information provider may migrate its services to other web-sites and purposely provide wrong information in the original site. However, this measure is drastic and difficult to realize. Firstly, the provider may not be directly affected by the beacon and hence does not have strong motivation to make the change. Secondly, migrating the services will also disrupt business activities, and providing wrong information affects the provider's credibility and may create legal issues.

6.2 Misleading the parser

If the actual program that accesses the beacon is made available, or in the context of DDoS, a copy of the agent is captured, then it can be analyzed for weaknesses. In particular, the preprogrammed parser that extracts the required information can be analyzed to find ways to mislead it, while keeping the content of the site unchanged. For example, it is possible that the parser may be unable to handle slight changes in reporting format, for instance, a change from "`DJIA 10427.20`" to "`DJIA 10,427.20`".

Another method is to craft the html page such that the preprogrammed parser will not only fail to obtain the required information, but obtain wrong information. For instance, placing false information in a commented section of the html page may mislead some parsers, but does not change the content presented to the human eye.

Since the above methods do not change the content of the web-pages, it may be easier to convince the service providers to collaborate in disrupting a particular way of beacon access. On the other hand, it is easy to improve the reliability of beacon access by simply using more than one website. Hence, many providers have to be convinced to implement the changes. This is not an easy task if numerous service providers are involved.

6.3 Using hard AI to disrupt the beacon

Instead of storing and displaying the indices explicitly, they can be stored and displayed in a form that is easily recognized by human, but not by current computer programs. This is similar to the use of hard AI [13] and graphical Turing test [9] in securing web-access, where the decimal figures are displayed as a spatially "warped" or "distorted" image.

Another effective method stores the actual indices in a transformed form, and use a script to reconstruct it. For example, the string may be stored in a reversed order and it is reconstructed during display. Note that the reconstruction script has to be made available to the public including the agents, and hence it is still possible to obtain the information. Nevertheless, the burden of program flow analysis is passed to the access program, who has to be generic enough to obtain the correct information.

Although the above two methods are effective, they generate overhead in network delay and processing, and may be unable to serve some legitimate users due to browser's compatibility or users who turn off certain browser's capabilities. With the use of hard AI, the distorted image may also appear strange to the users. Such inconveniences could turn away users. Hence, these methods are not desirable for providers in a competitive business environment.

7 Conclusion

In this paper, we describe a pervasive random beacon that is based on the closing indices of major stock markets. Such a random beacon meets the requirements of being random, unpredictable, is highly available and allows covert access. We demonstrate how such a random beacon can be constructed and present a use of the beacon for covert coordination of DDoS attack. Finally, we also present ways where the operation of beacon can be disrupted.

References

[1] Y. Aumann and M.O. Rabin. Information theoretically secure communication in the limited storage space model. *CRYPTO 1999*, pages 65–79, 1999.

[2] C.H. Bennett, D.P. DiVincenzo, , and R. Linsker. Digital recording system with time-bracketed authentication by on-line challenges and method for authenticating recordings. *US Patent 5764769*, 1998.

[3] Charles H. Bennett and John A. Smolin. Trust enhancement by multiple random beacons. *The Computing Research Repository(CoRR) cs.CR/0201003*, 2002. `http://xxx.lanl.gov/archive/cs/intro.html`.

[4] Drew Copley, Riley Hassell, Barnaby Jack, Karl Lynn, Ryan Permeh, and Derek Soeder. Blaster worm analysis. *eEye Digital Security*, 2003. `http://www.eeye.com/`.

[5] Donald E. Eastlake. Rfc 2777: Publicly verifiable nomcom random selection. *Internet RFC/STD/FYI/BCP Archives*, 2000.

[6] U.M. Maurer. Conditionally-perfect secrecy and a provably secure randomized cipher. *Journal of Cryptology*, 5:53–66, 1992.

[7] Ellen Messmer. Update: Blaster worm infections spreading rapidly. *Network World Fusion*, 2003. `http://www.nwfusion.com/`.

[8] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communications Review*, 34(2):39–54, 2004.

[9] William G. Morein, Angelos Stavrou, Debra L. Cook, Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. Using graphic turing tests to counter automated ddos attacks against web servers. *10th ACM Int. Conf. on Computer and Communications Security*, pages 8–19, 2003.

[10] Elchanan Mossel and Ryan O'Donnell. Coin flipping from a cosmic source: On error correction of truly random bits. To appear in Random Structures and Algorithms, 2004.

[11] Ryan Permeh and Marc Maiffret. ida "code red" worm analysis. *eEye Digital Security*, 2001. `http://www.eeye.com/`.

[12] M.O. Rabin. Transaction protection by beacons. *J. Computer and System Sciences*, 27(2):256–267, 1983.

[13] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. *EUROCRYPT*, 2003.

[14] John Walker. ent. *Fourmilab Switzerland*, 1998. `http://www.fourmilab.ch/`.