1

Internet Epidemics: Attacks, Detection and Defenses, and Trends

Zesheng Chen and Chao Chen
Department of Engineering, Indiana University - Purdue University Fort Wayne
Fort Wayne, IN 46805
USA

1. Introduction

Internet epidemics are malicious software that can self-propagate across the Internet, *i.e.,* compromise vulnerable hosts and use them to attack other victims. Since the early stage of the Internet, epidemics have caused enormous damages and been a significant security threat. For example, the Morris worm infected 10% of all hosts in the Internet in 1988; the Code Red worm compromised at least 359,000 hosts in one day in 2001; and the Storm botnet affected tens of millions of hosts in 2007. Therefore, it is imperative to understand and characterize the problem of Internet epidemics including the methods of attacks, the ways of detection and defenses, and the trends of future evolution.

Internet epidemics include viruses, worms, and bots. The past more than twenty years have witnessed the evolution of Internet epidemics. Viruses infect machines through exchanged emails or disks, and dominated 1980s and 1990s. Internet active worms compromise vulnerable hosts by automatically propagating through the Internet and have caused much attention since Code Red and Nimda worms in 2001. Botnets are zombie networks controlled by attackers through Internet relay chat (IRC) systems (*e.g.,* GTBot) or peer-to-peer (P2P) systems (*e.g.,* Storm) to execute coordinated attacks, and have become the number one threat to the Internet in recent years. Since Internet epidemics have evolved to become more and more virulent and stealthy, they have been identified as one of top four security problems and targeted to be eliminated before 2014 (52).

The task of protecting the Internet from epidemic attacks has many significant challenges:

– The original Internet architecture was designed without taking into consideration inherent security mechanisms, and current security approaches are based on a collection of "add-on" capabilities.

– New network applications and technologies become increasingly complex and expand constantly, suggesting that there will exist new vulnerabilities, such as zero-day exploits, in the foreseeable future.

– As shown by the evolution of Internet epidemics, attackers and the attacking code are becoming more and more sophisticated. On the other hand, the ordinary users cannot keep up with good security practices.

In this chapter, we survey and classify Internet epidemic attacks, detection and defenses, and trends, with an emphasis on Internet epidemic attacks. The remainder of this chapter is structured as follows. Section 2 proposes a taxonomy of Internet epidemic attacks. Section 3 discusses detection and defense systems against Internet epidemics. Section 4 predicts the trends of epidemic attacks. Finally, Section 5 concludes the paper.

2. Internet epidemic attacks

In this chapter, we focus on the self-propagation characteristic of epidemics, and use the terms “Internet epidemics” and “worms” interchangeably. A machine that can be compromised by the intrusion of a worm is called a *vulnerable host*, whereas a host that has been compromised by the attack of a worm is called an *infected host* or a *compromised host* or a *bot*. The way that a worm uses to find a target is called the *scanning method* or the *target discovery strategy*. *Worm propagation* is a procedure whereby a worm infects many hosts through Internet connections. In this section, we first identify three parameters that attackers can control to change the behavior of epidemic propagation. Next, we list the scanning methods that worms have used or will potentially exploit to recruit new bots and spread the epidemics. We also explain how these worm-scanning methods adjust the three parameters. Finally, we discuss the metrics that can be applied to evaluate worm propagation performance. The left of Figure 1 summarizes our taxonomy of Internet epidemic attacks.

2.1 Parameters controlled by worms

Three parameters that worms control to design the desired epidemic behaviors include

- *Scanning space:* the IP address space among which a worm searches for vulnerable hosts. A worm can scan an entire IPv4 address space, a routable address space, or only a subnetwork address space. Different bots may scan different address spaces at the same time.
- *Scanning rate:* the rate at which a worm sends out scans in the scanning space. A worm may dispatch as many scans as possible to recruit a certain number of bots in a short time or deliver scans slowly to behave stealthy and avoid detection.
- *Scanning probability:* the probability that a worm scans a specific address in the scanning space. A worm may use a uniform scanning method that hits each address in the scanning space equally likely or use a biased strategy that prefers scanning a certain range of IP addresses. Moreover, if the scanning probability is fixed at all time, the scanning strategy is called *static*; otherwise, the scanning probability varies with time, and the strategy is called *dynamic*.

All worm-scanning strategies have to consider these three parameters, adjusting them for different purposes (4). Although the parameters are local decisions that individual infected hosts make, they may lead to global effects on the Internet, such as the worm propagation speed, total malicious traffic, and difficulties in worm detection. In the following section, we demonstrate how different worm-scanning methods exploit these parameters.

2.2 Worm-scanning methods

Many worm-scanning methods have been used in reality or developed in the research community to spread epidemics. The methods include the following twelve representative strategies.

(1) *Random Scanning (RS)*

RS selects target IPv4 addresses uniformly (35; 6). Such a strategy is the simplest method and has been widely used by Internet worms such as Code Red (26), Slammer (25), and Witty (32). Specifically, RS probes the entire (*i.e.*, 2^{32}) IPv4 address space, uses a constant scanning rate,

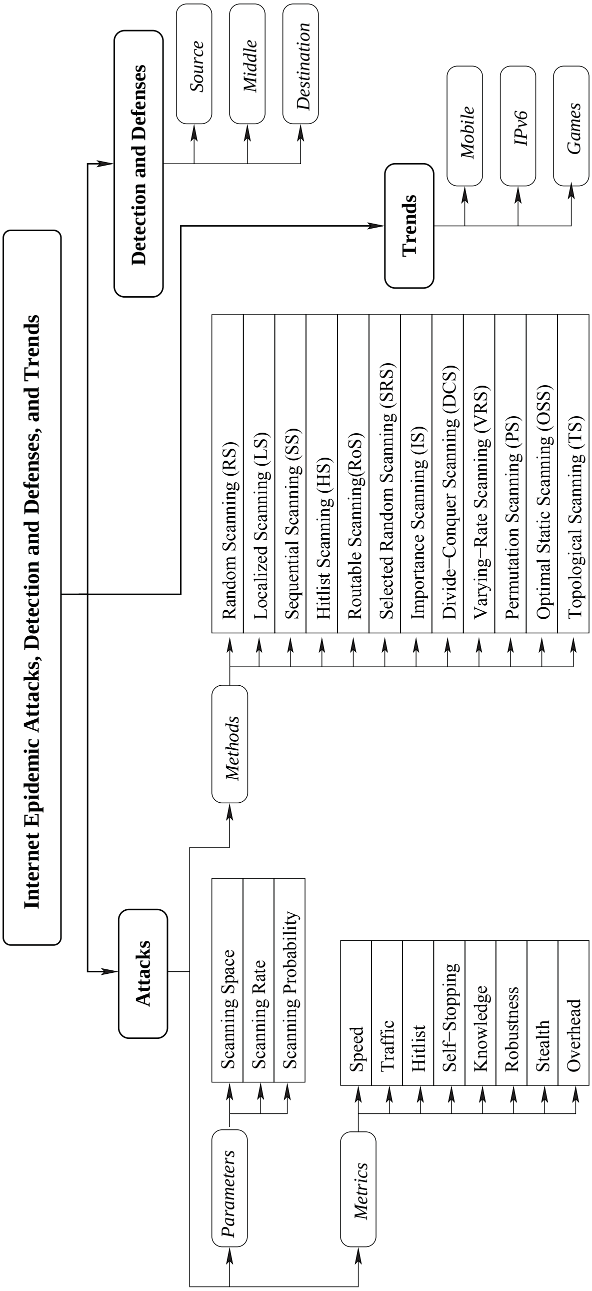

Fig. 1. A Taxonomy of Internet Epidemic Attacks, Detection and Defenses, and Trends.

and scans each address in the scanning space equally likely (*i.e.*, with the probability $1/2^{32}$).
(2) *Localized Scanning (LS)*
LS preferentially searches for targets in the "local" address space by designing the *scanning probability* parameter and has been used by such famous worms as Code Red II and Nimda (29; 5). For example, the Code Red II worm chooses a target IP address with the same first byte as the attacking machine with probability 0.5, chooses a target address with the same first two bytes with probability 0.375, and chooses a random address with probability 0.125. Similar to RS, LS probes the entire IPv4 address space and applies a constant scanning rate.
(3) *Sequential Scanning (SS)*
SS scans IP addresses sequentially from a randomly chosen starting IP address and has been exploited by the Blaster worm (49; 16; 10). Specifically, if SS is scanning address A now, it will continue to sequentially scan IP addresses $A+1$, $A+2$, $\cdots$ (or $A-1$, $A-2$, $\cdots$). Similar to RS, SS scans the entire IPv4 address space and uses a constant scanning rate. Although SS attempts to avoid re-scanning the IP addresses that have been probed, the scanning probability for SS can still be regarded as uniform. As a result, SS has a similar propagation speed as RS (49).
(4) *Hitlist Scanning (HS)*
HS collects a list of vulnerable hosts before a worm is released and attacks the hosts on the list first after the worm is set off (35; 40). Once the hosts on the list are compromised, the worm switches from HS to RS to infect the remaining vulnerable hosts. If the IP addresses of all vulnerable hosts are known to a worm in advance, HS leads to the fastest worm called the *flash worm* (34). Different from RS, HS only scans the hosts on the list before the list is exhausted. Moreover, HS is difficult to detect since each worm scan hits an existing host or service, which is indistinguishable from normal connections. But similar to RS, HS usually uses a constant scanning rate and selects targets on the list uniformly.
(5) *Routable Scanning (RoS)*
RoS scans only a routable address space (42; 50). According to the information provided by BGP routing tables, only about 28.6% of the entire IPv4 addresses are routable and can thus be used for real machines. Hence, RoS reduces the *scanning space* and spreads an epidemic much faster than RS. But similar to RS, RoS uses a constant scanning rate and selects targets in the routable address space uniformly.
(6) *Selected Random Scanning (SRS)*
Similar to RoS, SRS scans a partial IPv4 address space instead of the entire IPv4 address space (49; 31). For example, an attacker samples the Internet to detect an active IP address space before releasing a worm, and directs the worm to avoid scanning inactive addresses so that the worm can be stealthy for *network telescope* detection. Network telescopes use routable but unused IP addresses to detect worms and will be discussed in details in Section 3. Similarly, SRS applies a constant scanning rate and chooses targets in the scanning space uniformly.
(7) *Importance Scanning (IS)*
IS exploits the *scanning probability* parameter and probes different IP addresses with different probabilities (9; 8). Specifically, IS samples targets according to an underlying group distribution of vulnerable hosts. A key observation for IS is that vulnerable hosts distribute highly non-uniform in the Internet and form clusters (25; 26; 32; 29; 1; 10; 11; 38). Hence, IS concentrates on scanning groups that contain many vulnerable hosts to speed up the propagation. If a worm probes an IP address with probability 0, the worm would never scan this IP address. Therefore, RoS and SRS can be regarded as special cases of IS. Similarly, IS uses a constant scanning rate.

(8) *Divide-Conquer Scanning (DCS)*
DCS exploits the *scanning space* parameter, and different worm instances may probe different scanning spaces (42; 49; 4). Specifically, after an attacking host A infects a target B, A divides its scanning space into halves so that A would scan one half and B would scan the other half. As a result, the address space initially scanned by a worm will be partitioned into pieces that are probed by different infected hosts. Similar to RS, a worm instant uses a constant scanning rate and scans targets in its scanning space uniformly. In Section 2.3, however, it is demonstrated that DCS can spread an epidemic much faster than RS based on the realistic distribution of vulnerable hosts.

(9) *Varying-Rate Scanning (VRS)*
VRS varies the *scanning rate* over time to avoid detection (46; 47). Many worm detection methods have been developed based on change-point detection on the traffic going through routers or the unwanted traffic towards network telescopes. VRS, however, can potentially adjust its scanning rate dynamically so that it can smooth the malicious traffic. Similar to RS, VRS probes the IPv4 address space and scans targets in the scanning space uniformly.

(10) *Permutation Scanning (PS)*
PS allows all worm instances to share a common pseudo random permutation of the IP address space and to coordinate to provide comprehensive scanning (35). That is, the IPv4 address space is mapped into the permutation space, and an infected host uses SS in the permutation space. Moreover, if an infected host A hits another infected host B, A realizes that the scanning sequence starting from B in the permutation space has been probed and would switch to another scanning sequence to avoid duplicate scanning. In this way, compared with RS, PS can improve worm propagation performance (*i.e.,* the speed and the traffic) at the late stage. But at the early stage, PS behaves similar to RS in terms of the scanning space, the scanning rate, and the scanning probability.

(11) *Optimal Static Scanning (OSS)*
OSS minimizes the number of worm scans required to reach a predetermined fraction of vulnerable hosts by designing the proper *scanning probability* parameter (38). OSS is similar to IS since both methods exploit the scanning probability parameter. However, while IS emphasizes the speed of worm propagation, OSS focuses on the number of worm scans. In Section 2.3, we will further illustrate this point.

(12) *Topological Scanning (TS)*
TS exploits the information contained in the victim machines to locate new targets and has been used by Email viruses and Morris/SSH worms (40; 7). Hence, TS is a *topology-based* method, whereas the above eleven scanning strategies are *scan-based* methods. TS scans only neighbors on the topology, uses a constant scanning rate, and probes targets among neighbors uniformly.

2.3 Worm propagation performance metrics

How can we evaluate the performance of a worm-scanning method? In this section, we study several widely used performance metrics, focusing on scan-based epidemics.

(1) *Propagation Speed*
The epidemic propagation speed is the most used metric and defines how fast a worm can infect vulnerable hosts (35; 6; 49; 37; 36). Specifically, assume that two scanning methods A and B have the same initial conditions (*e.g.,* the number of vulnerable hosts and the scanning rate). If the numbers of infected hosts at time t for these two methods, $I_A(t)$ and $I_B(t)$, have the following relationship: $I_A(t) \geq I_B(t)$ for $\forall t \geq 0$, then method A has a higher propagation

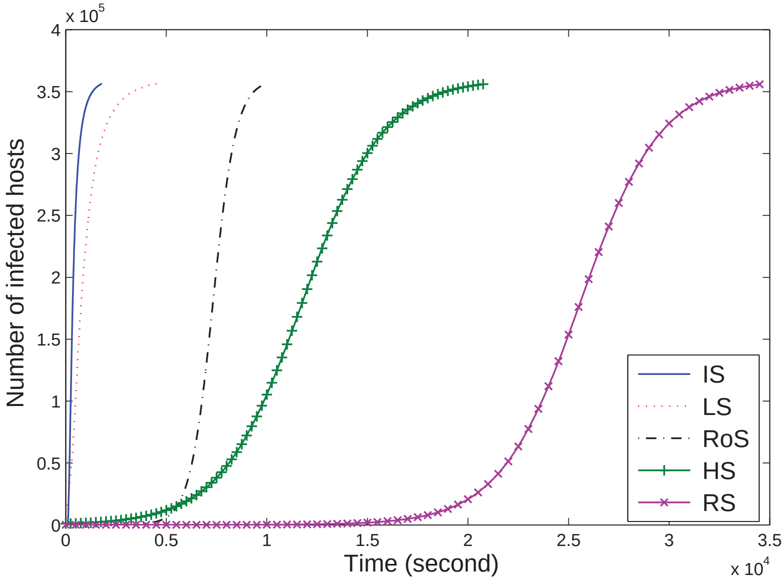

Fig. 2. Epidemic propagation speeds of different scanning methods (the vulnerable-host population is 360,000, the scanning rate is 358 per minute, the vulnerable-host distribution is from the DShield data with port 80, HS has a hitlist of 1,000, and other scanning methods start from an initially infected host).

speed than method *B*.

In Figure 2, we simulate a Code Red v2 worm using different scanning methods. Code Red v2 has a vulnerable-host population of 360,000 and a scanning rate of 358 per minute. To characterize scanning methods, we employ the analytical active worm propagation (AAWP) model and its extensions (6). The AAWP model applies a discrete-time mathematical difference equation to describe the spread of RS and has been extended to model the propagation of other advanced scanning methods. In Figure 2, we compare IS, LS, RoS, and HS with RS. We assume that except HS, a worm begins spreading from an initially infected host. HS has a hitlist size of 1,000. Since the Code Red v2 worm attacks Web servers, we use the DShield data (54) with port 80 as the distribution of vulnerable hosts. DShield collects intrusion detection system and firewall logs from the global Internet (54; 1; 11). We also assume that once a vulnerable host is infected, it will stay infected. From the figure, it is seen that IS, LS, RoS, and HS can spread an epidemic much faster than RS. Specifically, it takes RS 10 hours to infect 99% of vulnerable hosts, whereas HS uses only about 6 hours. RoS and LS can further reduce the time to 3 hours and 1 hour. IS spreads fastest and takes only 0.5 hour.

The design of most advanced scanning methods (*e.g.,* IS, LS, RoS, and OSS) roots on the fact that vulnerable hosts are not uniform distributed, but highly clustered (9; 29; 49; 38). Specifically, the Internet is partitioned into sub-networks or groups according to such standards as the first byte of IP addresses (/8 subnets), the IP prefix, autonomous systems, or DNS top-level domains. Since the distribution of vulnerable hosts over groups is highly uneven, a worm would avoid scanning groups that contain no or few vulnerable hosts and concentrate on scanning groups that have many vulnerable hosts to increase the propagation

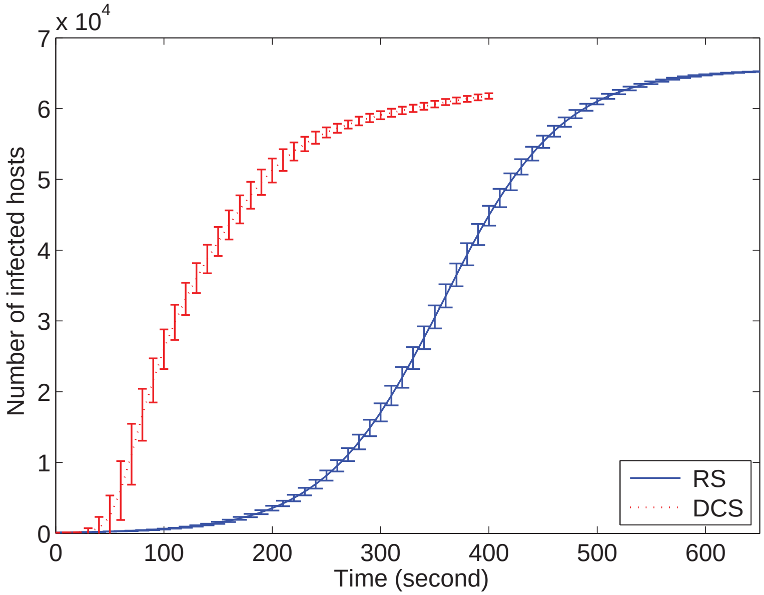

Fig. 3. Comparison of DCS and RS (the vulnerable-host population is 65,536, the scanning rate is 1,200 per minute, the vulnerable-host distribution follows that of Witty-worm victims, and a hitlist size is 100).

speed. Moreover, once a vulnerable host in a sub-network with many vulnerable hosts is infected, a LS worm can rapidly compromise all the other local vulnerable hosts (29; 5).
DCS is another scanning method that exploits the highly uneven distribution of vulnerable hosts, but has been studied little (4). Imagine a toy example where vulnerable hosts only distribute among the first half of the IPv4 address space and no vulnerable hosts exist in the second half of the space. A DCS worm starts from an initially infected host, which behaves like RS until hitting a target. After that, the initially infected host scans the first half of the space, whereas the new bot probes the other half. While the new bot cannot recruit any target, the initially infected host would find the vulnerable hosts faster with the reduced scanning space. This fast recruitment in the first half of the space would in return accelerate the infection process since the newly infected hosts in the area only scan the first half of the space. In some sense, DCS could lead an epidemic to spread towards an area with many vulnerable hosts. Figure 3 compares DCS with RS, using a discrete event simulator. The simulator implements each worm scan through a random number generator and simulates each scenario with 100 runs using different seeds. The curves represent the mean of 100 runs, whereas the error bars show the variation over 100 runs. The worm has a vulnerable population of 65,536, a scanning rate of 1,200 per second, and a hitlist size of 100. The distribution of vulnerable hosts follows that of Witty-worm victims provided by CAIDA (56). Figure 3 demonstrates that DCS spreads an epidemic much faster than RS. Specifically, RS takes 479 seconds to infect 90% of vulnerable hosts, whereas DCS takes only 300 seconds.
(2) *Worm Traffic*
Worm traffic is defined as the total number of worm scans (38). Specifically, assuming that a worm uses a constant scanning rate s and infects $I(t)$ machines at time t, we can approximate

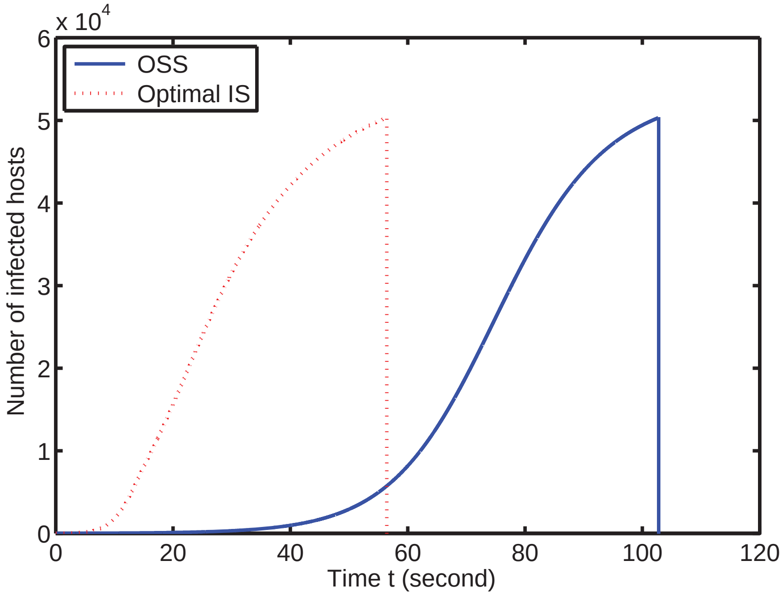

Fig. 4. Comparison of OSS and optimal IS (the vulnerable-host population is 55,909, the scanning rate is 1,200 per minute, the vulnerable-host distribution follows that of Witty-worm victims, and a hitlist size is 10).

worm traffic by time t as $s \cdot \int_0^t I(x)dx$. An epidemic may intend to reduce the worm traffic to elude detection or avoid too much scanning traffic that would slow down worm propagation in return. OSS is designed to minimize the traffic required to reach a predetermined fraction of vulnerable hosts (38).

The two metrics, the propagation speed and the worm traffic, reflect different aspects of epidemics and may not correlate. For example, two scanning methods can use the same number of worm scans to infect the same number of vulnerable hosts, but differ significantly on the propagation speed. Specifically, we apply the extensions of the AAWP model to characterize the spread of OSS and optimal IS, as shown in Figure 4. Here, we simulate the propagation of the Witty worm, where the vulnerable-host population is 55,909, the scanning rate is 1,200 per minute, the vulnerable-host distribution follows that of Witty-worm victims, and a hitlist size is 10. Both scanning methods use 1.76×10^9 worm scans to infect 90% of vulnerable hosts (*i.e.*, the scanning rate multiples the area under the curve). However, OSS uses 102 seconds to infect 90% vulnerable hosts, whereas optimal IS takes only 56 seconds.

(3) *Initially Infected Hosts (Hitlist)*

A hitlist defines the hosts that are infected at the beginning of worm propagation and reflects the attacks' ability in preparing the worm attacks (35). The curves of HS and RS in Figure 2 show that a worm can spread much faster with a larger hitlist. Hence, an attacker may use a botnet (*i.e.*, a network of bots) as a hitlist to send out worm infection (14). Moreover, the locations of the hitlist affect LS. For example, if the hitlist resides in sub-networks with few vulnerable hosts, the worm cannot spread fast at the early stage.

(4) *Self-Stopping*

If a worm can self-stop after it infects all or most vulnerable hosts, it can reduce the chance to

be detected and organize the network of bots in a more stealthy way (23). One way for a bot to know the saturation of infected hosts is that it has hit other bots for several times. Another way is that a worm estimates the number of vulnerable hosts and the scanning rate, and thus predicts the time to compromise most vulnerable hosts.
(5) *Knowledge*
The use of knowledge by an attacker can help a worm speed up the propagation or reduce the traffic (8; 38). For example, IS exploits the knowledge of the vulnerable-host distribution, assuming that this distribution is either obtainable or available. Based on the knowledge, worm-scanning methods can be classified into three categories:

– *Blind:* A worm has no knowledge about vulnerable hosts and has to use oblivious scanning methods such as RS, LS, SS, and DCS.

– *Partial:* A scanning strategy exploits partial knowledge about vulnerable hosts, such as RoS, SRS, IS, and OSS.

– *Complete:* A worm has the complete knowledge about vulnerable hosts, such as a flash worm (34).

A future intelligent worm can potentially learn certain knowledge about vulnerable hosts while propagating. Specifically, a blind worm uses RS to spread and collect the information on vulnerable hosts at the very early stage, and then switches to other advanced scanning methods (*e.g.,* SRS, IS, or OSS) after estimating the underlying distribution of vulnerable hosts accurately. We call such worms *self-learning worms* (8).
(6) *Robustness*
Robustness defines a worm's ability against bot failures. For example, DCS is not robust since the failure of a bot at the early stage may lead to the consequence that a worm misses a certain range of IP addresses (4). Therefore, redundancy in probing the same scanning space may be necessary to increase the robustness of DCS. Comparatively, RS, SS, RoS, IS, PS, and OSS are robust since except extreme cases (*e.g.,* all initially infected hosts fail before recruiting a new bot), a small portion of bot failures do not affect worm infection significantly.
(7) *Stealth*
Stealth defines a worm's ability in avoiding detection. For example, many worm detection methods root on change-point detection on the unwanted traffic towards network telescopes or the traffic going through routers (51; 48; 2). These methods, however, may fail to detect VRS that adjusts the worm traffic to spread an epidemic under the radar (46; 47). Another stealthy scanning method is HS that makes worm infection undistinguishable from normal connections (35).
(8) *Overhead*
Overhead defines the size of additional packet contents required for a worm to design a scanning method. For example, the flash worm may require a very large storage to contain the IP addresses of all vulnerable hosts (34). Specifically, if there are 100,000 vulnerable hosts, the flash worm demands 400,000 bytes to store the IP addresses without compression. Such large overhead slows down the worm propagation speed and introduces extra worm traffic.

3. Internet epidemic detection and defenses

To counteract notorious epidemics, many detection and defense methods have been studied in recent years. Based on the location of detectors, we classify these methods into the following three categories. The top-right of Figure 1 summarizes our taxonomy of Internet epidemic detection and defenses.

3.1 Source detection and defenses

Source detection and defenses are deployed at the local networks, protecting local hosts and locating local infected hosts (17; 18; 41; 36; 19). For example, a defense system applies the latest patches to end systems so that these systems can be immunized to epidemic attacks that exploit known vulnerabilities. To detect infected hosts, researchers have characterized epidemic host behaviors to distinguish them from the normal host behaviors. For example, an infected host attempts to spread an epidemic as quickly as possible and sends out many scans to different destinations at the same time. Comparatively, a normal host usually does not connect to many hosts simultaneously. Hence, a detection and defense system can explore this difference and build up a connection queue with a small length (*e.g.*, 5) for an end host. Once the queue is filled up, the further connection request would be rejected. In this way, the spread of an epidemic is slowed down, while the normal hosts are affected little. Moreover, monitoring the queue length can reveal the potential appearance of a worm. Such a method is called *virus throttling* (36). Another detection method targets the inherent feature of scan-based epidemics. Specifically, since a bot does not know the (exact) locations of vulnerable hosts, it guesses the IP addresses of targets, which leads to the likely failures of connections and differs from normal connections. A *sequential hypothesis testing* method has been proposed to exploit such a difference and shown to identify an RS bot quickly (17; 18; 41).

3.2 Middle detection and defenses

Middle detection and defenses are deployed at the routers, analyzing the on-going traffic and filtering out the malicious traffic (27; 43; 33; 21). *Content filtering* and *address blacklisting* are two commonly used techniques (27). Content filtering uses the known signatures to detect and remove the attacking traffic, whereas address blacklisting filters out the traffic from known bots. Similar to source detection and defenses, middle detection and defenses can also explore the inherent behaviors of epidemics and differ the malicious traffic from the normal traffic. For example, several sampling methods have been proposed to detect the super spreader – a host sends traffic to many hosts, and thus identify potential bots (43). Another method is based on the distributions of source IP addresses, destination IP addresses, source port numbers, and destination port numbers, which would change after a worm is released (33; 21).

3.3 Destination detection and defenses

Destination detection and defenses are deployed at the *Darknet* or *network telescopes*, a globally routable address space where no active servers or services reside (51; 53; 55). Hence, most traffic arriving at Darknet is malicious or unwanted. CAIDA has used a /8 sub-network as network telescopes and observed several large-scale Internet epidemic attacks such as Code Red (26), Slammer (25), and Witty (32) worms.

We coin the term *Internet worm tomography* as inferring the characteristics of Internet epidemics from the Darknet observations (39), as illustrated in Figure 5. Since most worms use scan-based methods and have to guess target IP addresses, Darknet can observe partial scans from bots. Hence, we can combine Darknet observations with the worm propagation model and the statistical model to detect the worm appearance (42; 2) and infer the worm characteristics (*e.g.,* the number of infected hosts (6), the propagation speed (48), and the worm infection sequence (30; 39)). Internet worm tomography is named after *network tomography*, where end system observations are used to infer the characteristics of the internal network (*e.g.,* the link delay, the link loss rate, and the topology) (3; 12). The common approach to network tomography is to formulate the problem as a linear inverse problem. Internet

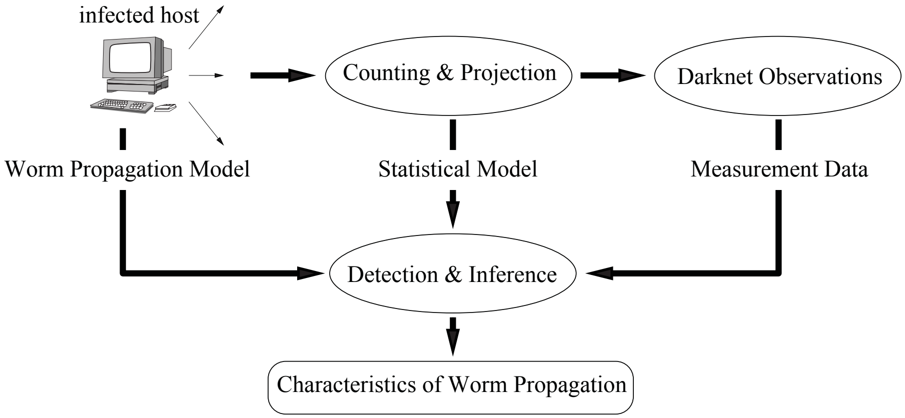

Fig. 5. Internet Worm Tomography (39).

worm tomography, however, cannot be translated into the linear inverse problem due to the complexity of epidemic spreading, and therefore presents new challenges. Several statistical detection and estimation techniques have been applied to Internet worm tomography, such as maximum likelihood estimation (39), Kalman filter estimation (48), and change-point detection (2).

Figure 6 further illustrates an example of Internet worm tomography on estimating when a host gets infected, *i.e.*, the host infection time, from our previous work (39). Specifically, a host is infected at time instant t_0. The Darknet monitors a portion of the IPv4 address space and can receive some scans from the host. The time instants when scans hit the Darknet are t_1, t_2, $\cdots$, t_n, where n is the number of scans received by the Darknet. Given Darknet observations t_1, t_2, $\cdots$, t_n, we then attempt to infer t_0 by applying advanced estimation techniques such as maximum likelihood estimation.

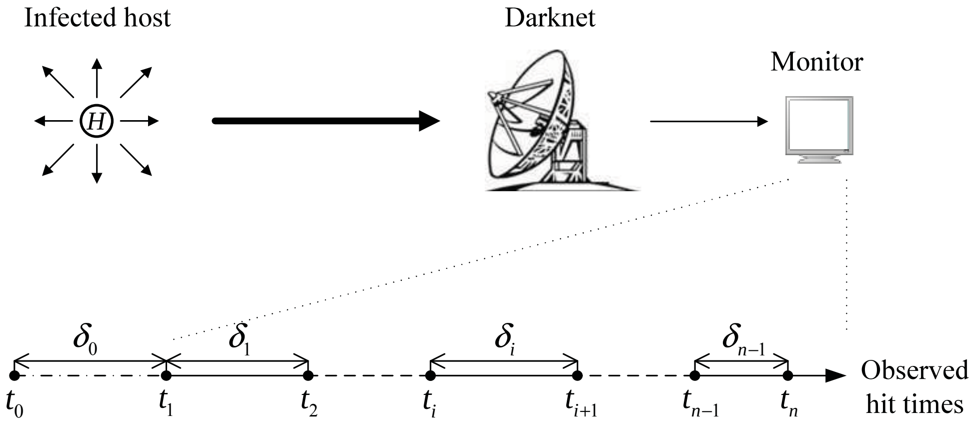

Fig. 6. An illustration of Darknet observations (39).

4. Internet epidemic trends

Internet epidemics have evolved in the past more than twenty years and will continue developing in the future. In this section, we discuss three prominent trends of epidemic attacks. The bottom-right of Figure 1 summarizes our taxonomy of Internet epidemic trends.

4.1 Mobile epidemics

Over the past few years, a new type of worms has emerged that specifically targets portable devices such as cell phones, PDAs, and laptops. These mobile worms can use Internet connectivity for their propagation. But more importantly, they can apply TS and spread directly from device to device, using a short-range wireless communication technology such as WiFi or Bluetooth (20; 44). The first mobile epidemic, Cabir, appeared in 2004 and used Bluetooth channels on cell phones running the Symbian operation system to spread onto other phones. As WiFi/Bluetooth devices become increasing popular and wireless networks become an important integrated part of the Internet, it is predicted that epidemic attacks will soon become pervasive among mobile devices, which strongly connect to our everyday lives.

4.2 IPv6 worms

IPv6 is the future of the Internet. IPv6 can increase the scanning space significantly, and therefore, it is very difficult for an RS worm to find a target among the 2^{128} IP address space (50). The future epidemics, however, can still spread relatively fast in the IPv6 Internet. For example, we find that if vulnerable hosts are still clustered in IPv6, an IS worm can be a zero-day worm (10). Moreover, a TS epidemic can spread by exploiting the topological information, similar to Morris and SSH worms. Another example of advanced worms would propagate by guessing DNS names in IPv6, instead of IP addresses (15).

4.3 Propagation games

To react to worm attacks, a promising method generates self-certifying alerts (SCAs) or patches from detected bots or known vulnerabilities and uses an overlay network for broadcasting SCAs or patches (13; 37). A key factor for this method to be effective is indeed that SCAs or patches can be disseminated much faster than worm propagation. This introduces propagation games between attackers and defenders, since both sides apply epidemic spreading techniques. Such a weapon race would continue in the foreseeable future.

5. Conclusions

In this chapter, we have surveyed a variety of techniques that Internet epidemics have used or will potentially exploit to locate targets in the Internet. We have examined and classified existing mechanisms against epidemic attacks. We have also predicted the coming threats of future epidemics.

In addition to survey, we have compared different worm scanning methods based on the three important worm-propagation parameters and different performance metrics. Specifically, we have demonstrated that many advanced scanning methods can spread a worm much faster than random scanning. Moreover, the worm propagation speed and the worm traffic reflect different aspects of Internet epidemics and may not correlate. We have also emphasized Internet worm tomography as a framework to infer the characteristics of Internet epidemics from Darknet observations. Finally, we have contemplated that epidemics can spread among mobile devices and in IPv6, and have a far-reaching effect to our everyday lives.

6. References

[1] P. Barford, R. Nowak, R. Willett, and V. Yegneswaran, "Toward a model for sources of Internet background radiation," in *Proc. of the Passive and Active Measurement Conference (PAM'06)*, Mar. 2006.

[2] T. Bu, A. Chen, S. V. Wiel, and T. Woo, "Design and evaluation of a fast and robust worm detection algorithm," in *Proc. of INFOCOM'06*, Barcelona, Spain, April 2006.

[3] R. Caceres, N.G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Transactions on Information Theory*, vol. 45, no. 7, Nov. 1999, pp. 2462-2480.

[4] C. Chen, Z. Chen, and Y. Li, "Characterizing and defending against divide-conquer-scanning worms," *Computer Networks*, vol. 54, no. 18, Dec. 2010, pp. 3210-3222.

[5] Z. Chen, C. Chen, and C. Ji, "Understanding localized-scanning worms," in *Proc. of 26th IEEE International Performance Computing and Communications Conference (IPCCC'07)*, New Orleans, LA, Apr. 2007, pp. 186-193.

[6] Z. Chen, L. Gao, and K. Kwiat, "Modeling the spread of active worms," in *Proc. of INFOCOM'03*, vol. 3, San Francisco, CA, Apr. 2003, pp. 1890-1900.

[7] Z. Chen and C. Ji, "Spatial-temporal modeling of malware propagation in networks," *IEEE Transactions on Neural Networks: Special Issue on Adaptive Learning Systems in Communication Networks*, vol. 16, no. 5, Sept. 2005, pp. 1291-1303.

[8] Z. Chen and C. Ji, "A self-learning worm using importance scanning," in *Proc. ACM/CCS Workshop on Rapid Malcode (WORM'05)*, Fairfax, VA, Nov. 2005, pp. 22-29.

[9] Z. Chen and C. Ji, "Optimal worm-scanning method using vulnerable-host distributions," *International Journal of Security and Networks: Special Issue on Computer and Network Security*, vol. 2, no. 1/2, 2007.

[10] Z. Chen and C. Ji, "An information-theoretic view of network-aware malware attacks," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, Sept. 2009, pp. 530-541.

[11] Z. Chen, C. Ji, and P. Barford, "Spatial-temporal characteristics of Internet malicious sources," in *Proc. of INFOCOM'08 Mini-Conference*, Phoenix, AZ, Apr. 2008.

[12] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet Tomography," *IEEE Signal Processing Magazine*, May 2002, pp. 47-65.

[13] M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham, "Vigilante: End-to-end containment of Internet worms,", in Proc. of SOSP'05, Brighton, UK, Oct. 2005.

[14] D. Dagon, C. C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proc. 13th Annual Network and Distributed System Security Symposium (NDSS'06)*, San Diego, CA, Feb. 2006.

[15] H. Feng, A. Kamra, V. Misra, and A. D. Keromytis, "The effect of DNS delays on worm propagation in an IPv6 Internet," in *Proc. of INFOCOM'05*, vol. 4, Miami, FL, Mar. 2005, pp. 2405-2414.

[16] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, and G. Riley, "Worm detection, early warning and response based on local victim information," in *Proc. 20th Ann. Computer Security Applications Conf. (ACSAC'04)*, Tucson, AZ, Dec. 2004.

[17] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," in *Proc. of IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.

[18] J. Jung, S. Schechter, and A. Berger, “Fast detection of scanning worm infections,” in *7th International Symposium on Recent Advances in Intrusion Detection (RAID’04)*, Sophia Antipolis, French Riviera, France, Sept. 2004

[19] S. A. Khayam, H. Radha, and D. Loguinov, “Worm detection at network endpoints using information-theoretic traffic perturbations,” in *Proc. of IEEE International Conference on Communications (ICC’08)*, Beijing, China, May 2008.

[20] J. Kleinberg, “The wireless epidemic,” *Nature (News and Views)*, vol. 449, Sept. 2007, pp. 287-288.

[21] A. Lakhina, M. Crovella, and C. Diot, “Mining anomalies using traffic feature distributions,” in *Proc. of ACM SIGCOMM’05*, Philadelphia, PA, Aug. 2005.

[22] M. Lelarge and J. Bolot, “Network externalities and the deployment of security features and protocols in the Internet,” in *Proc. of the 2008 ACM SIGMETRICS*, June 2008, pp. 37-48.

[23] J. Ma, G. M. Voelker, and S. Savage, “Self-stopping worms,” in *Proc. ACM/CCS Workshop on Rapid Malcode (WORM’05)*, Fairfax, VA, Nov. 2005, pp. 12-21.

[24] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attacks and defense mechanisms,” *ACM SIGCOMM Computer Communications Review*, vol. 34, no. 2, April 2004, pp. 39-54.

[25] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, “Inside the Slammer worm,” *IEEE Security and Privacy*, vol. 1, no. 4, July 2003, pp. 33-39.

[26] D. Moore, C. Shannon, and J. Brown, “Code-Red: a case study on the spread and victims of an Internet worm,” in *ACM SIGCOMM/USENIX Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[27] D. Moore, C. Shannon, G. Voelker, and S. Savage, “Internet quarantine: Requirements for containing self-propagating code,” in *Proc. of INFOCOM’03*, vol. 3, San Francisco, CA, Apr., 2003, pp. 1901-1910.

[28] J. Nazario, *Defense and Detection Strategies Against Internet Worms*. Artech House, Inc., Norwood, MA, 2003.

[29] M. A. Rajab, F. Monrose, and A. Terzis, “On the effectiveness of distributed worm monitoring,” in *Proc. of the 14th USENIX Security Symposium (Security’05)*, Baltimore, MD, Aug. 2005, pp. 225-237.

[30] M. A. Rajab, F. Monrose, and A. Terzis, “Worm evolution tracking via timing analysis,” in *Proc. ACM/CCS Workshop on Rapid Malcode (WORM’05)*, Fairfax, VA, Nov. 2005, pp. 52-59.

[31] M. A. Rajab, F. Monrose, and A. Terzis, “Fast and evasive attacks: highlighting the challenges ahead,” in *Proc. of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID’06)*, Hamburg, Germany, Sept. 2006.

[32] C. Shannon and D. Moore, “The spread of the Witty worm,” *IEEE Security and Privacy*, vol. 2, no 4, Jul-Aug 2004, pp. 46-50.

[33] S. Singh, C. Estan, G. Varghese, and S. Savage, “Automated worm fingerprinting,” in *Proc. of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI’04)*, San Francisco, CA, Dec. 2004, pp. 45-60.

[34] S. Staniford, D. Moore, V. Paxson, and N. Weaver, “The top speed of flash worms,” in *Proc. ACM/CCS Workshop on Rapid Malcode (WORM’04)*, Washington DC, Oct. 2004, pp. 33-42.

[35] S. Staniford, V. Paxson, and N. Weaver, “How to 0wn the Internet in your spare time,” in *Proc. of the 11th USENIX Security Symposium (Security’02)*, San Francisco, CA, Aug. 2002, pp. 149-167.

[36] J. Twycross and M. M. Williamson, "Implementing and testing a virus throttle," in *Proc. of the 12th USENIX Security Symposium (Security'03)*, Washington, DC, Aug. 2003, pp. 285-294.

[37] M. Vojnovic and A. J. Ganesh, "On the race of worms, alerts and patches," *IEEE/ACM Transactions on Networking*, vol. 16 , no. 5, Oct. 2008, pp. 1066-1079.

[38] M. Vojnovic, V. Gupta, T. Karagiannis, and C. Gkantsidis, "Sampling strategies for epidemic-style information dissemination," in *Proc. of INFOCOM'08*, Phoenix, AZ, April 2008, pp. 1678-1686.

[39] Q. Wang, Z. Chen, K. Makki, N. Pissinou, and C. Chen, "Inferring Internet worm temporal characteristics," in *Proc. IEEE GLOBECOM'08*, New Orleans, LA, Dec. 2008.

[40] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A taxonomy of computer worms," in *Proc. of ACM CCS Workshop on Rapid Malcode*, Oct. 2003, pp. 11-18.

[41] N. Weaver, S. Staniford, and V. Paxson, "Very fast containment of scanning worms," in *Proc. of 13th Usenix Security Conference (Security'04)*, San Diego, CA, Aug. 2004.

[42] J. Xia, S. Vangala, J. Wu, L. Gao, and K. Kwiat, "Effective worm detection for various scan techniques," *Journal of Computer Security*, vol. 14, no. 4, 2006, pp. 359-387.

[43] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang, "Worm origin identification using random moonwalks," in *Proc. of the IEEE Symposium on Security and Privacy (Oakland'05)*,Oakland, CA, May 2005.

[44] G. Yan and S. Eidenbenz, "Modeling propagation dynamics of bluetooth worms (extended version)," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, March 2009, pp. 353-368.

[45] V. Yegneswaran, P. Barford, and D. Plonka, "On the design and utility of internet sinks for network abuse monitoring," in *Symposium on Recent Advances in Intrusion Detection (RAID'04)*, Sept. 2004.

[46] W. Yu, X. Wang, D. Xuan, and D. Lee, "Effective detection of active smart worms with varying scan rate," in *Proc. of IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks (SecureComm'06)*, Aug. 2006.

[47] W. Yu, X. Wang, D. Xuan, and W. Zhao, "On detecting camouflaging worm," in *Proc. of Annual Computer Security Applications Conference (ACSAC'06)*, Dec. 2006.

[48] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The monitoring and early detection of Internet worms," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, Oct. 2005, pp. 961-974.

[49] C. C. Zou, D. Towsley, and W. Gong, "On the performance of Internet worm scanning strategies," *Elsevier Journal of Performance Evaluation*, vol. 63. no. 7, July 2006, pp. 700-723.

[50] C. C. Zou, D. Towsley, W. Gong, and S. Cai, "Advanced routing worm and its security challenges," *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 82, no. 1, 2006, pp.75-85.

[51] CAIDA, "Network telescope," [Online]. Available: http://www.caida.org/research/security/telescope/ (Aug./2010 accessed).

[52] Computing Research Association, "Grand research challenges in information security & assurance," [Online]. Available: http://archive.cra.org /Activities /grand.challenges /security /home.html (Aug./2010 accessed).

[53] Darknet. [Online]. Available: http://www.cymru.com/Darknet/. (Oct./2010 accessed).

[54] Distributed Intrusion Detection System (DShield), http://www.dshield.org/.

(Oct./2010 accessed).

[55] Honeypots: Tracking Hackers. [Online]. Available: http : // www. tracking - hackers. com/ . (Oct./2010 accessed).

[56] The CAIDA Dataset on the Witty Worm - March 19-24, 2004, Colleen Shannon and David Moore, http://www.caida.org/data/passive/witty_worm_dataset.xml. Support for the Witty Worm Dataset and the UCSD Network Telescope are provided by Cisco Systems, Limelight Networks, the US Department of Homeland Security, the National Science Foundation, DARPA, Digital Envoy, and CAIDA Members.