The Ecology of Malware

Jedidiah R. Crandall
University of New Mexico
Dept. of Computer Science
Mail stop: MSC01 1130
1 University of New Mexico
Albuquerque, NM 87131-0001
crandall@cs.unm.edu

Roya Ensafi
University of New Mexico
Dept. of Computer Science
Mail stop: MSC01 1130
1 University of New Mexico
Albuquerque, NM 87131-0001
royaen@cs.unm.edu

Stephanie Forrest
University of New Mexico
Dept. of Computer Science
Mail stop: MSC01 1130
1 University of New Mexico
Albuquerque, NM 87131-0001
forrest@cs.unm.edu

Joshua Ladau
Santa Fe Institute
1399 Hyde Park Road
Santa Fe, New Mexico 87501
jladau@santafe.edu

Bilal Shebaro
University of New Mexico
Dept. of Computer Science
Mail stop: MSC01 1130
1 University of New Mexico
Albuquerque, NM 87131-0001
bshebaro@cs.unm.edu

ABSTRACT

The fight against malicious software (or *malware*, which includes everything from worms to viruses to botnets) is often viewed as an "arms race." Conventional wisdom is that we must continually "raise the bar" for the malware creators. However, the multitude of malware has itself evolved into a complex environment, and properties not unlike those of ecological systems have begun to emerge. This may include competition between malware, facilitation, parasitism, predation, and density-dependent population regulation. Ecological principles will likely be useful for understanding the effects of these ecological interactions, for example, carrying capacity, species-time and species-area relationships, the unified neutral theory of biodiversity, and the theory of island biogeography. The emerging malware ecology can be viewed as a critical challenge to all aspects of malware defense, including collection, triage, analysis, intelligence estimates, detection, mitigation, and forensics. It can also be viewed as an opportunity.

In this position paper, we argue that taking an ecological approach to malware defense will suggest new defenses. In particular, we can exploit the fact that interactions of malware with its environment, and with other malware, are neither fully predictable nor fully controllable by the malware author—yet the emergent behavior will follow general ecological principles that can be exploited for malware defense.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection–*invasive software*

NSPW'08, September 22–25, 2008, Lake Tahoe, California, USA.

General Terms

Security

Keywords

malware ecology, malware analysis, worms, viruses, botnets

1. INTRODUCTION

Modern malware defense involves a variety of activities and is quickly becoming unsustainable. So many malware samples are collected from the wild each day that triaging is necessary to determine which samples warrant further analysis. Much of the analysis framework is consumed by creating new signatures for new variants of well-known attacks, but simply releasing a signature for every threat that is currently in the wild is only part of what defenders must do. They must also aggregate the results of multiple analyses to mitigate threats that have already spread, develop intelligence estimates of what the attackers are doing and are likely to do next, and conduct forensic analysis of attacks that have occurred.

The study of how ecological principles can be applied to these defense activities is important for two reasons. First, the emerging malware ecology is straining defenses at every stage. Exactly identifying every distinct instance of malicious code that is in the wild, something that most malware defense relies on, is no longer possible. For example, it is now common for targeted attacks, *e.g.*, phishing attacks against local credit unions where e-mails are sent only to e-mail accounts in a particular city, to be constructed using tools for building advanced Trojan horses using a point-and-click interface, such as Shark 2 [14]. As a second example, Song *et al.* [32] documented both experimentally and with information theory arguments that the diversity of a single component (the decoder) of a botnet greatly exceeds the capacity of any signature-based detection algorithm.

The second reason to study malware ecology is that it has the potential of leading to defenses that give fundamental advantages to the defender. Current approaches are aimed primarily at known attacks or known types of attack. In the current landscape, these approaches at best consume some time of the attacker, who must

create new malware to evade the defenses. Before the malware landscape became so ecologically diverse, it might have taken 30 minutes to roll out a new defense in response to the latest attack, and two days for a new attack to appear in response to that defense (these amounts of time are estimates for the purpose of example only). However, with the advent of "malware 2.0" [4], malware is supported on the back end by large amounts of resources and coordination, and it might take a couple of hours to roll out a new defense but only minutes for a new attack to appear. With the possible exception of automated diversity techniques, anti-malware techniques are so predictable that there is little defenders can to do to prevent sophisticated attackers from gaining this advantage. This is because the *game* between attacker and defender is circumscribed within the limits of the controlled environments in which analysis and defense are carried out. We posit that there is an important difference between the richly diverse interactions that malware has with its real environment in the wild, and the more restricted behaviors of malware in clean and controlled laboratory environments. Studying malware ecology has the potential to bring the complexity of the environment, something that the attacker cannot predict or control, within the scope of novel analysis and defense techniques.

1.1 Today's analysis environments

Malware analysis is typically conducted in carefully controlled environments. A typical malware defense pipeline must sift through tens of thousands of unique malware samples every day to find the few that are truly novel and warrant further analysis. As a rule, these selected samples are traded among experts and analyzed thoroughly only if their behavior can be demonstrated in a reproducible way on a clean, freshly installed copy of the vulnerable system. This prevents wasted effort, when one malware analyzer passes malware off to another. If a sample "misbehaves" in some way, the malware analyzer will sift through all similar samples to find one that behaves more predictably (the behavior can be reproduced). Subsequent efforts at forensics and threat estimation take these manual analyses as their starting point. At every stage valuable information is lost because we do not know how to use this information effectively. In the wild, malware that behaves exactly as its author intended is the exception, not the rule [20].

Suppose that ecologists studied only healthy organisms in isolation, in pristine caged environments tailored to those organisms. All of their efforts to estimate the impacts of introduced species, to predict populations of pests, and to explain changes in the wild populations of endangered species would be hampered. Malware analysis professionals are aware of this problem but lack alternatives. For example, Peter Szor, one of the most respected malware practitioners in the industry, describes [34, Section 9.8] the large number of interactions, both accidental and intentional, between various malware instances. He describes how worms that spread over the Internet by copying a `*.exe` file from machine to machine often carry three or four file infector viruses "on their back." This can cause "mutant" worms whose signature changes polymorphically even though the worm itself is not polymorphic. It can also cause large resurgences of virulent file infector viruses that are no longer on the threat list. As Szor states, "anti-virus programs need to address this issue." Many file infector viruses, such as the infamous Chernobyl virus, were important only because of their ability to spread on the backs of e-mail worms. To date such malware-malware synergies have been mostly accidental, although there are intentional examples as well, such as the CTX virus' strategy of piggybacking on the Cholera worm. However, it is only a matter of time before malware authors realize the potential of this technique.

1.2 The emerging malware ecology

In addition to interactions with each other, today's malware interacts with its environment in unprecedented ways. The Trojan programs installed by advanced botnets, dubbed "malware 2.0" by some researchers [4], use techniques such as combining public-key cryptography with existing peer-to-peer networks to create custom name resolution mechanisms. There is also a growing trend of corrupting the existing Domain Name Service (DNS) system for command and control [12], creating the rise of a "second secret authority." Additionally, botnets such as Storm cannot be described as a single instance of malicious code, they are really a system of different Trojans, amorphous and heterogeneous command and control, drive-by downloaders, droppers, waves of malicious e-mails that recruit new victims, and any number of other components that are nothing new in isolation but taken together represent a new level of threat on the Internet. Obtaining a "sample" of the Storm botnet and analyzing it in a laboratory is thus complicated in fundamental ways. Unless the laboratory environment contains unwitting peer-to-peer users and corrupted DNS resolution paths, it is incomplete.

Even the relatively simple malware of the past was notoriously fragile, and it was difficult to execute in a virtual machine because of the many environmental dependencies. Most simple e-mail worms, for example, exit abnormally when executed in an isolated virtual machine because, *e.g.*, the system date is not set within the right range, DNS queries go unanswered, or no SMTP server is configured. As early as 2003 researchers reported that running vulnerable programs under an emulator was sufficient to disrupt some attacks, even without patching the vulnerability [3, 5, 21]. This is a serious problem because it requires malware analyzers to fully recreate every detail of the malware's natural environment before analysis can be performed, and the complexity of malware environments continues to rise.

1.3 Opportunities

The emerging malware ecology is also troublesome from the attacker's point of view. In fact, it always has been. Suppose an attacker wants to write a worm to install a backdoor on every machine it infects. They develop a zero-day exploit, use their own custom polymorphic engine, and throw all of their newest and best tricks into the code to slow the antivirus analysis down. They also use stealth techniques that they know will allow it to spread unfettered for at least a day or two. It is the Perfect Worm.

After they release it, within a couple of days a dozen "script kiddies" have grabbed samples of it and made slight changes to the code, mostly to install their own backdoor, and then released their variants into the wild. Thus, the original author's variant is relegated to the name PerfectWorm.D (if by chance it was the fourth variant to be found in the wild). Furthermore, when they log into infected machines through their backdoor, they find that these unpatched machines typically are rebooted every so often by an old worm that is still making the rounds, the NotSoElegant Worm, which makes its presence conspicuous in other ways, too. Then somebody releases the PerfectWormKiller Worm, which exploits a buffer overflow they accidentally coded in their backdoor to get into infected machines and remove the PerfectWorm.

None of this is uncommon. For example, consider the variants of major worms, most of which are just trivial changes to, for example, the date of the payload and the URL used for a DoS attack. The Cabir worm, the first worm for cell phones to be seen in the wild, became Cabir.A, Cabir.AA, Cabir.AB, Cabir.AC, Cabir.AD, and so on, Cabir.B, Cabir.BA, Cabir.C, Cabir.D, and so on to Cabir.Z. The Creeper worm was possibly the first computer worm ever, appearing in the early 70's, and was immediately targeted for removal by

the Reaper worm which followed in its tracks. Code Red was attacked by Code Green, and Blaster was attacked by Welchia. Sasser was both attacked by Gaobot.AJS, which came in through the same vulnerability as Sasser and then used a *vampire attack*[1] to disable it, and targetted by Dabber, which exploited a buffer overflow in Sasser's crude FTP server to spread. Doomjuice spread using MyDoom's backdoor. (For details about all of these examples, see `symantec.com` and Szor's book [34]).

1.4 Structure of the paper

The next section, Section 2, gives examples of how ecological principles can be applied to malware defense, and lists some related works. Section 3 presents some initial results that demonstrate that the distribution of malware in a honeypot network trace is affected by competition between malware. This is followed by Section 4 where we discuss how an ecologically inspired approach to computer security can give defenders an inherent advantage in the "arms race," and contrast our proposed approach with biologically inspired approaches. Then we describe the discussion that occurred at the NSPW workshop along with our reflections on it in Section 5, and conclude.

2. EXPLOITING ECOLOGICAL PRINCIPLES

This section gives some examples of how ecological principles can be applied to malware defense, and discusses some related work in computer science.

2.1 Malware defense using ecological principles

First, consider malware detection of known threats (*e.g.*, such as those detected by an antivirus scanner, although these ideas could apply to anomaly detection as well). A key question is: What is the most efficient detection scheme? The malware defense community uses the concept of a "wild list" to refer to the set of viruses that are currently prevalent in production computers and networks and therefore are worth scanning for. Scanning for only the viruses on the wild list is an optimization that avoids the overhead of searching for every possible threat that has ever been observed. Principles of ecology could be used to generalize this idea by introducing the notion of indicator species [15] to distill the wild list down to a small list of indicator viruses. An indicator species is a species that is known to be correlated with the presence of one or more other species. An example of why this might occur is nestedness [1], where the least common species are observed to almost always occur on the same islands as common species, creating a nested structure of occurrence frequencies.

Ideas such as nestedness and indicator species also apply to malware collection and mitigation. To achieve this, it is critical to know where malware can be found, which means we must know how malware distributes itself over space and time. Island biogeography [27] and related ideas, such as the species-area relationship [9] and species-time relationship [30], can provide quantitative analysis tools for this purpose. Ecologists have many tools for understanding these phenomena and routinely exploit them to benefit the biodiversity. As one example of questions ecologists might pose and then answer, consider the problem of taking a habitat and finding the optimal subset of the habitat that should be set aside to preserve the biodiversity (*e.g.*, if only a certain amount of acreage can be dedicated to preservation efforts). A similar question could be posed for malware ecologies, except that the goal would be to eliminate as many species as possible using only limited resources for defense.

Triage and analysis are important components of the malware defense pipeline. Here, an understanding of ecological principles will not only be beneficial in the near future, but absolutely necessary. Triaging is the process of sifting through a large amount of collected samples to decide which warrant further analysis, and analysis is the process of detailing what a particular sample does and possibly generating a signature for it. Complex interactions between malware in the wild, including parasitism, predation, facilitation, and commensalism, will be even more prevalent in the future. We need to understand these important interactions. Commensalism, for example, could mean that a file infector virus spreads by attaching itself to a circulating e-mail worm. If such unclean samples are routinely removed from the analysis pipeline, then analyzing the virus in isolation may not reveal important behaviors of the virus when combined with the worm. As an example, the worm might set a registry entry to load a copy of itself every time the host machine is rebooted, therefore putting the virus in the same machine startup path. If the worm and virus are separated during analysis then, in later stages of analysis, potentially important diurnal patterns in the virus' actual behavior might be missed. Many of the underlying principles for such interactions are well understood in the field of community ecology [29].

Each of these ecological principles becomes important if the ultimate goal is to understand malware ecology as a whole rather than as a single instance. This is relevant for intelligence estimates about the state of malware in the wild and forensics for law enforcement or other purposes. When combining data from several telescopes and honeypots, for example, many factors play a role in the way that the data should be aggregated, including competition, habitat filtering, predator-prey dynamics [35], and all of the aforementioned principles from biogeography and community ecology.

2.2 Related work

Although no coordinated effort exists to promote the inclusion of ecological principles throughout the malware defense pipeline, a few related works are worth mentioning here. Dancho Danchev has discussed the "malware ecology" in white papers [13] and on his blog [14]. In this paper, we take "malware ecology" to mean malware's interactions with its environment and with other malware, Danchev also includes social processes and other factors into the environment when using the term. An interesting application of ecological methods and principles to a computer science problem was Weaver and Collins' use of capture-recapture models to estimate the extent of phishing activity on the Internet [36]. Also of interest is the general idea of combining analysis of a malware sample with analysis of its environment for powerful approaches to analysis [10] and forensics [24]. The dynamics of direct competition between worms has been explored [7]. Interesting examples about how malware interacts with its environment and with other malware in unexpected ways can be found in Szor's book [34]. Finally, the difference between how malware is meant to behave by its authors and how it actually behaves in the wild [20] is enlightening.

3. EXAMPLE: COMPETITION

In this section, we present some initial results demonstrating competition between different "species" of malware in the wild. To assess these effects, we use a null modeling approach developed in ecology.

[1]Vampire attack is a technical term referring to an attack that hijacks control flow by placing jumps throughout memory.

3.1 Constructing a presence-absence matrix

We used network trace data from the Minos project [11]. This is honeypot data collected over a period of two years at a single honeypot location. Many null models used by ecologists take as input a binary presence-absence matrix that has species as its rows and sites as its columns, with a 1 indicating the presence of a species at that site and a 0 otherwise. Additionally, for the method that we used, which is due to Ladau and Schwager [26], the species must be grouped into "units" (e.g., genera, families, functional groups) reflecting ecological similarity. This is because Ladau and Schwager's method is based on a sampling process that models competition as a reduction in the co-occurrence of ecologically similar species. This requires an *a priori* definition of which species are ecologically similar and therefore likely to compete, but has the benefit of not depending on unjustifiable parametric assumptions.

For the initial results presented here, we consider the destination port number of an observed attack to indicate species. A probe on a particular port from a remote IP address implies that the remote IP address is infected with a worm that spreads using that particular port. For example, an observed SYN packet from IP address $w.x.y.z$ on port 135 in the Minos network trace data is taken to mean that $w.x.y.z$ is infected with the species corresponding to port 135, *i.e.*, the Blaster worm.

The power (or sensitivity) of the analysis in Ladau and Schwager's method requires an ample number of species per site. We consider a site to be a $/16$ subnetwork, so a distinct site covers any of the $65,536$ IP addresses in $w.x.?.?$. Using $/16$ subnetworks is necessary because if sites are defined as single IP addresses or $/24$ subnetworks, then individual sites have too few species for the analysis to have power. For example, for $/24$ subnetworks with 256 IP addresses to each site, only 20 sites have four or more species. Thus, our choice of $/16$ subnetworks is largely due to the characteristics of the Minos dataset. As for larger sites, we did not use $/8$ subnetworks because there are only a small handful of Class A networks where considering them as one administrative entity is meaningful.

Using $/16$ subnetworks also removes concerns about network address translation, because a site is no longer a single machine. Our results show that competition occurs even at this higher level of abstraction, for reasons explained in Section 3.3. We filtered out any obvious port scanning and fingerprinting traffic before constructing the presence-absence matrix, because this traffic does not indicate the presence of any particular malware. The end result is that there are $1,453$ distinct sites in our presence-absence matrix.

The eight "species" are grouped into the following units: Ports 135 and 445 are grouped into the "consumes unpatched Windows machines using remote memory corruption" unit; ports 137 and 139 are grouped into the "consumes unpatched Windows machines using unprotected file shares" unit; ports 80 and 443 are grouped into the "targets web servers" unit; and ports 1433 and 1434 are grouped into the "targets SQL servers" unit.

Port 135 probes typically indicate a Blaster variant. Note that Welchia, a predator of Blaster that used the same vulnerability as Blaster to spread on port 135, was programmed to self-terminate on 1 July 2004, so is not prevalent in the Minos data except in early traces where it is indistinguishable from Blaster due to our definition of species. Port 445 indicates either variants of the Sasser worm or botnet activity. Ports 137 and 139 indicate a variety of worms that spread through network shares including Sadmind, Chode, and Qaz. Port 443 probably indicates an unpatched Windows machine in a botnet that is scanning for web servers, while port 80 can either indicate the same or could be a worm such as Code Red that spreads from web server to web server. The same is true of ports 1433 and 1434, with the former probably indicating botnet activity and the latter being overwhelmingly due to the Slammer worm.

One possible concern is that the content filtering performed by certain ISPs to reduce traffic from particular worms would introduce biases into our measurements. Since we consider only SYN probes this effect applies only to the slammer worm which spread as a single UDP packet.

3.2 Hypothesis tests for competition, facilitation, and habitat filtering

After constructing the presence-absence matrix, we used null model tests [26] to check for effects of competition, facilitation (*i.e.*, positive interactions between species), and habitat filtering (*i.e.*, differences in site accessibility and habitability) on the co-occurrence patterns of the malware. Hypothesis testing allows us to postulate a null hypothesis, *i.e.*, that effects of a particular process (*e.g.*, competition) are not present in the data, and then test that hypothesis. If we reject the null hypothesis, then it is unlikely that data generated from the null model (*i.e.*, without the effect in question) could have generated the data, modulo a false positive rate that can be calculated as a test diagnostic. If we fail to reject the null hypothesis, then the test is considered inconclusive. In this case, another test diagnostic, the power of the test (which is 1 minus the false negative rate), gives some indication of the meaning of the result.

Many null model tests for competition, facilitation, and habitat filtering have been developed in ecology [8, 17, 18, 33], but they often require assumptions that each species is equally likely to occur at all sites, or that at each site every species is equally likely to occur [25]. These assumptions are not satisfied in the Minos honeypot data set.

The fact that every $/16$ subnetwork has a different make-up of Windows machines, web servers, and SQL servers, combined with the fact that each machine is vulnerable to a distinct set of species, violates both assumptions. For example, if one subnetwork has many SQL servers and only a single HTTP server, and another subnetwork has only one SQL server and many HTTP servers, then: 1) Slammer is much more likely to occur at the former site than at the latter (and conversely for Code Red); and 2) At the latter site Code Red is more likely to occur than Slammer (and conversely for the former site). Note that the presence of a particular type of server does not necessarily imply a 100% probability that the server is vulnerable, so it is not necessarily the case that any servers at all of a given type implies that that subnetwork will eventually be infected with the corresponding worm.

Recently developed tests [26] require only one parametric assumption, which is met by the Minos data. Let i and j denote the ith and jth species of malware to arrive at a site (a site is a $/16$ subnetwork for our present purposes), respectively, and $\langle i, j \rangle$ the event that i and j belong to the same unit (see above). The tests require the assumption that $P(\langle i, j \rangle) > 0$ for all i and j. Since there is nothing about worms in either unit that precludes worms of the same unit also entering an infected subnetwork this assumption is sound for the Minos data. Another key assumption is site independence (*i.e.*, the species observed at a given site do not affect which species are observed at another site), which is met by the Minos data. While random IP address scanning worms and botnets tend to prefer closer subnets by using random IP probing strategies that prefer the $/16$ and $/8$ subnetworks close to them (by mostly sticking to their own $/24$ or $/16$ ranges), over time this effect averages out. Thus, independence between sites is maintained if the time scale is long enough.

The tests check for independence during the colonization process: at a given site, if malware interact, then the presence of certain species should make other species less (competition) or more (facilitation) likely to occur. We employed two tests [26]: a test that can detect effects of competition (the "competition test"), and a test that can detect effects of facilitation and habitat filtering (the "facilitation test"). For further details, see Ladau and Schwager [26].

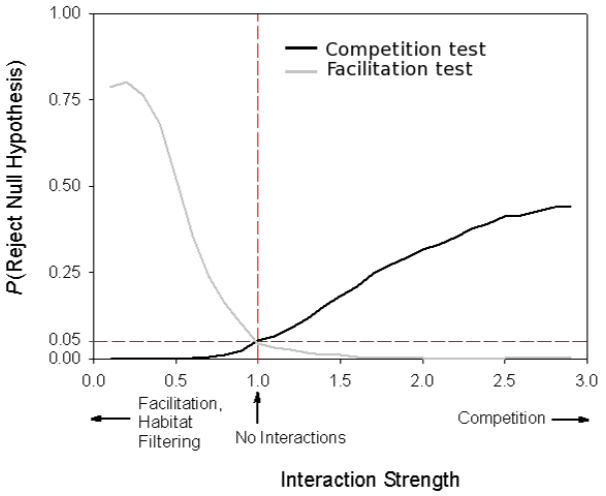

Figure 1: Diagnostic criteria for the null model tests used here. This shows that both of our conclusions, about the existence of competition and the lack of habitat filtering in the Minos dataset, are based on statistical methods with low false positive and false negative rates. The interaction factor is a parameter in a model of community assembly which reflects the strength and type of interactions between species (Ladau and Schwager, in preparation). The null hypotheses for the competition and facilitation null model tests state that the interaction factor is less than and greater than 1, respectively. For the competition test, a false positive occurs when the interaction factor is 1 or less (left of the vertical line) and the test rejects the null hypothesis. A false negative occurs when the interaction factor is greater than 1 (right of the vertical line) and the test fails to reject the null hypothesis. Thus, the fact that the curve for the facilitation test is low over the interval $[0, 1]$ indicates controlled false positive rates, and the fact that it is relatively high over the interval $(1, 3]$ indicates low false negative rates. For the facilitation test, the respective intervals are reversed.

The results of hypothesis testing on the Minos dataset are as follows. The competition test had a p-value (labeled *Pr(Reject Null Hypothesis)* on the y-axis of Figure 1) of less than 0.0001, providing strong evidence of competition (the p-value is the probability that the null model could have produced data at least as extreme as the observed data). Test diagnostics showed a Type I error rate (or false positive rate) of 0.055. This indicates that the significant result was likely indeed due to competition affecting community assembly, rather than a false positive result. By contrast, the facilitation test had a high p-value (≈ 1), indicating that there was no evidence for inferring effects of habitat filtering or facilitation. In addition, the facilitation test had high power (*i.e.*, sensitivity), shown in Figure 1, with a maximum of 0.802, indicating that had there been such effects the test would likely have detected them. Overall, these results suggest that the distribution of the malware is affected by competition, but unaffected by habitat filtering and facilitation.

Since it is well-known that web server worms tend to inhabit web server sites, and SQL server worms tend to inhabit SQL server sites, why was no habitat filtering observed? Habitat filtering means that the presence of one member of a unit makes it more likely that another member of the same unit will appear in that habitat, because something about the habitat is more conducive to species in that unit. The null hypothesis was not rejected for the habitat filtering/facilitation test because infected machines scanning on ports 443 or 1443 are likely to not be actual web or SQL servers, but simply infected Windows machines in a botnet that are scanning for servers. This fact does not significantly affect the result that competition is present, the null hypothesis for the competition test is rejected without including web servers or SQL servers in the data (data not shown).

3.3 Explanation for observed competition

The competition observed in the Minos data set is largely between port 135 and port 445 (this can be determined by excluding particular species and re-executing the tests). Here we give two plausible explanations for this. Both explanations stem from the fact that the Blaster worm uses an exploit that conspicuously reboots the machine upon infection. The Blaster worm also makes it possible for reinfection, by either itself or another worm, to occur using the same exploit. Thus a machine vulnerable to the Blaster worm will be rebooted repeatedly, often several times a day. Recall that the observed competition is at the $/16$ subnetwork level of abstraction, which captures both single-machine effects, such as installing patches, and higher-level effects, such as testing a system administrator's patience to the point that they decide to, *e.g.*, firewall all Windows file sharing ports or install an intrusion detection system.

One explanation for competition between these two species is that bot herders (attackers who control the bots in a botnet) commonly install a patch for the port 135-accessible vulnerability that Blaster variants use. They do this to make the machine more stable and its infected state less conspicuous to the machine's owner. Since the majority of botnets during the time-period in which the Minos network traces were recorded used the same exploit as the Sasser worm, on port 445, to spread, it would be expected that machines vulnerable on this port would eventually become invulnerable on port 135.

Another explanation is that the system administrators of networks where many Windows machines are vulnerable to Blaster (on port 135) were alerted to the vulnerability of their networks because of the conspicuous nature of Blaster and its variants. Thus they firewalled all Windows file sharing ports, installed intrusion detection, became more vigilant about installing patches, *etc.* In making machines invulnerable to port 135 attacks, they also made them invulnerable to port 445 attacks. This can also go the other way when botnet-infected machines are used in denial-of-service attacks and SPAM campaigns, making port 445 attacks the conspicuous species in this case. In other words, the resource that both port 135 and port 445 species consume that leads to competition between these two species is the *patience* of the people that control machines on the subnetwork.

4. ECOLOGY AND THE "ARMS RACE"

How can ecology ideas and techniques give us as the defenders an advantage in our arms race with attackers? In this section we compare the proposed ecologically inspired approach for com-

puter security research with biologically inspired approaches, discuss how exploiting emergent patterns of malware can give us as the defenders a natural advantage, and give a hypothetical example to illustrate this.

4.1 Ecologically inspired vs. biologically inspired approaches

Our ecological approach to computer security research is biologically inspired in that it highlights certain biological properties of software installations. It differs from most earlier work in the following ways: 1) rather than mimicking the architecture and mechanisms of natural systems and devising algorithms with similar properties, it studies the organizational structure and interactions among components using quantitative analysis methods that were developed for studying biological systems; 2) It focuses on all aspects of malware defense, including collection, triage, analysis, intelligence estimates, detection, mitigation, and forensics, whereas most earlier work employing biological approaches has emphasized detection and mitigation. It is important to consider the entire process for developing effective malware defenses.

Biologically inspired approaches typically borrow analogies from the methods that biological systems use to achieve certain properties (such as detecting intrusions or failing gracefully), and then translate these analogies into computational methods. Regardless of whether we have learned all we should from biology, have only just begun to scratch the surface, or have borrowed too much (see last year's NSPW panel [31]), the emphasis has been on translating biological systems into computational mechanisms. Ecology is different from biology in that it examines how various organisms and their environment interrelate. We posit that the interrelations between biological systems and the interrelations between computational systems are relatively similar when compared to the similarity of the inner workings of these two kinds of systems. Since ecology focuses on interrelations and biology focuses on inner workings, the former is more directly applicable than the latter. In other words, the statistical methods and underlying ideas that ecologists have developed can be directly applied to malware defense.

The NSPW panel last year classified biologically inspired efforts into three broad categories: computer immune systems, diversity, and autonomic computing. All three of these categories focus on detection, mitigation, or both. Throughout this paper we have given examples of how ecologically inspired approaches could apply to all of the aspects of malware defense that we identify. It is worth noting that an important area of previous work which does not fall into the three categories identified by the panel is the extensive work on epidemiology of computer viruses [2, 19, 22, 23, 28]. This work is relevant to the malware ecology approach because it uses quantitative analysis methods and it considers properties of an entire system.

4.2 Exploiting emergent patterns

In other defense scenarios, *e.g.*, military strategy, defenders sometimes place their defenses in certain positions to take advantage of physical characteristics, such as mountains or rivers. The placements are designed to to put the enemy at a disadvantage during an attack. This is how we envision ecological approaches to computer security: because of their uncertainty about how their malware will interact with the environment and other malware, ecological patterns will emerge that the attacker can neither predict nor control. Our longterm goal is to use ideas and statistical techniques from ecology to identify and exploit such patterns.

4.3 A hypothetical example

Consider, as an example, turnover rates in insular biogeography [6]. Ecologists have observed that the number of species on any given island stays relatively constant, but the rate at which species go extinct and are replaced by new species, called the turnover rate, depends on several factors including the distance from the source of immigration. This is because distance has an effect on both the immigration rate and extinction rate of the island. If this emergent behavior were also observed among malware such as botnets on the Internet, then measuring the turnover rate for some definition of island (*e.g.*, a subnetwork) could be used to identify the sources of immigration (*i.e.*, where the bot Trojans are first appearing). This would be useful for collection, intelligence estimates, mitigation, and forensics.

5. DISCUSSION AT NSPW

The discussion at the NSPW workshop centered around two related themes: low-level analogies between ecology and malware that will allow us to directly apply methods from the latter to the former, and higher-level analogies between these two fields that would provide fundamental insights.

We argued that because both malware and ecology would be termed "organized complexity" in Gerald Weinberg's *General Systems Thinking* [37], a lot of the analysis methods that ecologists have developed over the years could be directly applied to malware defense (see Section 2.1 for examples). In the small group discussion it was pointed out that many other fields, *e.g.*, economics and theoretical physics, have problem domains that are too organized for purely statistical analysis and too complex for mechanical analysis. It was also pointed out that these basic ideas could be applied to software engineering in general and not just to malware. We gratefully acknowledge these points, but chose to preserve the scope of this paper as insights from ecology applied to malware defense because of the compelling analogies between living organisms and malicious code.

Much of the plenary discussion centered around how to define "randomness" in describing a system as being organized complexity. The main idea is that certain events are unpredictable to an observer, which we colloquially termed "randomness" as in Weinberg's famous graph [37, Page 18]. As an example, in thermodynamics the observer's lack of knowledge about the locations of individual gas particles can be quantified eloquently as entropy to give concise equations for analysis [16, Chapter 5], and in classical physics forces, inertia, angles, and so forth can be quantified directly into mechanical equations. Our point was that for systems with organized complexity neither of these techniques, the purely statistical or the purely mechanical, allow us to fully analyze a system. Because ecologists have a long history of analyzing these types of systems, and because malware has certain analogies with ecology, malware analysis can borrow many powerful techniques from ecology. The exact definition of "randomness" is orthogonal to this point, but perhaps "unpredictability" would have been a better word to use.

Another discussion, which we only scratched the surface of at this year's NSPW, was whether the analogy between ecological systems and malware ran deeper than just techniques that could be borrowed from one field and applied to another. In analogies between the immune system and computer security, fundamental insights such as the importance of diversity and the distinction between self and non-self have had transformative effects on computer security research. We believe that if the analogy between malware and ecology is to provide these kinds of deeper insights

then a fundamental question about the analogy must be addressed: what is is that malware consumes and metabolizes? Metabolism is a key concept in ecology, and is viewed by some as a unifying principle [38]. An analogous concept for malicious code could lead to fundamental insights, but what is it that malware consumes? Resources such as CPU cycles and network bandwidth? The attention of the human beings that use the computers? Dollars? We leave this question for future work.

6. CONCLUSION

This position paper argued that the ecology of malware is an important area for the research community to study and will lead to more effective defenses. Two major points in our argument were that: 1) Ecologically-inspired defense techniques can give defenders an inherent advantage in the "arms race" with attackers; and 2) Ideas and statistical techniques from ecology can be directly applied to all aspects of malware defense, rather than simply serve as analogies for detection and mitigation. We gave examples of how ecological principles could be applied to malware defense, and presented evidence of competition between malware as a preliminary result. It is our hope that this position paper will engender discussion about malware ecology.

7. ACKNOWLEDGMENTS

Stephanie Forrest gratefully acknowledges the support of the National Science Foundation (grants CCF–0621900 and CCR–0331580), Air Force Office of Scientific Research (MURI grant FA9550–07–1–0532), and the Santa Fe Institute. Jedidiah Crandall greatly appreciates being supported through MURI grant FA9550–07–1–0532. Joshua Ladau was funded by a postdoctoral fellowship from the Santa Fe Institute. We would also like to thank our shepherd, Michael Locasto, the anonymous NSPW reviewers, the members of our small group session at NSPW, and the NSPW attendees for their valuable insights. This work also benefited from many others who have discussed these ideas with us including Melanie Moses and James Brown.

8. REFERENCES

[1] W. Atmar and B. D. Patterson. The measure of order and disorder in the distribution of species in fragmented habitat. *Oecologia*, 96(3):373–382, December 1993.

[2] J. Balthrop, S. Forrest, M. Newman, and M. Williamson. Technological networks and the spread of computer viruses. *Science*, 304:527–9, 2004.

[3] G. Barrantes, D. Ackley, S. Forrest, T. Palmer, D. Stefanovic, and D. Zovi. Randomized instruction set emulation to disrupt binary code injection attacks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, 2003.

[4] K. Baumgartner. Storm 2007: Malware 2.0 has arrived. Virus Bulletin, 2007.

[5] S. Bhatkar, D. C. DuVarney, and R. Sekar. Address obfuscation: an efficient approach to combat a broad range of memory error exploits. In *12th USENIX Security Symposium*, Washington, DC, August 2003.

[6] J. H. Brown and A. Kodric-Brown. Turnover Rates in Insular Biogeography: Effect of Immigration on Extinction. *Ecology*, 58(2):445–449, 1977.

[7] F. Castaneda, E. C. Sezer, and J. Xu. Worm vs. worm: preliminary study of an active counter-attack mechanism. In *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode*, pages 83–93. ACM Press, 2004.

[8] E. Connor and D. Simberloff. The Assembly of Species Communities: Chance or Competition? *Ecology*, 60(6):1132–1140, 1979.

[9] E. F. Connor and E. D. McCoy. The statistics and biology of the species-area relationship. *The American Naturalist*, 113(6):791–883, June 1979.

[10] J. R. Crandall, G. Wassermann, D. A. S. de Oliveira, Z. Su, S. F. Wu, and F. T. Chong. Temporal search: detecting hidden malware timebombs with virtual machines. In *ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pages 25–36, New York, NY, USA, 2006. ACM Press.

[11] J. R. Crandall, S. F. Wu, and F. T. Chong. Minos: Architectural support for protecting control data. *ACM Trans. Archit. Code Optim.*, 3(4):359–389, 2006.

[12] D. Dagon, N. Provos, C. P. Lee, and W. Lee. Corrupted DNS resolution paths: The rise of a malicious resolution authority. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS 2008)*, Feb. 2008.

[13] D. Danchev. Malware: future trends (white paper), 2006.

[14] D. Danchev. Mindstreams of information security knowledge (`http://ddanchev.blogspot.com/`), 2008.

[15] M. Dufrene and P. Legendre. Species assemblages and indicator species: The need for a flexible asymmetrical approach. *Ecological Monographs*, 67(3):345–366, 1997.

[16] R. P. Feynman. *Feynman Lectures on Computation*. Perseus Books, Cambridge, MA, USA, 2000.

[17] B. Fox. Species assembly and the evolution of community structure. *Evolutionary Ecology*, 1(3):201–213, 1987.

[18] M. Gilpin and J. Diamond. Factors contributing to non-randomness in species Co-occurrences on Islands. *Oecologia*, 52(1):75–84, 1982.

[19] K. J. Hall. Thwarting network stealth worms in computer networks through biological epidemiology. Ph.D. Thesis, Virginia Polytechnic Institute and State University, 2006.

[20] John Canavan. Me code write good: The l33t skillz of the virus writer (Symantec White Paper).

[21] G. S. Kc, A. D. Keromytis, and V. Prevelakis. Countering code-injection attacks with instruction-set randomization. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 272–280, New York, NY, USA, 2003. ACM.

[22] J. O. Kephart, G. B. Sorkin, W. C. Arnold, D. M. Chess, G. J. Tesauro, and S. R. White. Biologically inspired defenses against computer viruses. In *IJCAI '95*. International Joint Conference on Artificial Intelligence, 1995.

[23] J. O. Kephart, S. R. White, and D. M. Chess. Computers and epidemiology. *IEEE Spectrum*, 30(5):20–26, 1993.

[24] A. Kumar, V. Paxson, and N. Weaver. Exploiting underlying structure for detailed reconstruction of an internet-scale event. In *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2006. ACM Press.

[25] J. Ladau. Validation of null model tests using Neyman-Pearson hypothesis testing theory. *Theoretical Ecology*, In Press, 2008.

[26] J. Ladau and S. Schwager. Robust Hypothesis Tests for Independence in Community Assembly. *Journal of Mathematical Biology*, 57:537–555, 2008.

[27] R. H. MacArthur and E. O. Wilson. *The Theory of Island Biogeography*, volume 1 of *Monographs in Population Biology*. Princeton Press, 1967.

[28] M. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(035101), 2002.

[29] E. P. Odum. Fundamentals of ecology. W. B. Saunders Co., Philadelphia and London. 546 p., 1959.

[30] M. Rosenzweig. Preston's ergodic conjecture: the accumulation of species in space and time. *Biodiversity Dynamics*, July 1998.

[31] A. Somayaji, M. Locasto, and J. Feyereisl. Panel: The Future of Biologically-Inspired Security: Is There Anything Left to Learn? In the Proceedings of the 2007 New Security Paradigms Workshop.

[32] Y. Song, M. E. Locasto, A. Stavrou, A. D. Keromytis, and S. J. Stolfo. On the infeasibility of modeling polymorphic shellcode. In *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 541–551, New York, NY, USA, 2007. ACM.

[33] L. Stone and A. Roberts. The checkerboard score and species distributions. *Oecologia*, 85:74–79, 1990.

[34] P. Szor. *The Art of Computer Virus Research and Defense*. Symantec Press, 2005.

[35] V. Volterra. Variations and fluctuations of the number of individuals in animal species living together. *Animal Ecology*, August 1931.

[36] R. Weaver and M. P. Collins. Fishing for phishes: applying capture-recapture methods to estimate phishing populations. In *eCrime '07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 14–25, New York, NY, USA, 2007. ACM.

[37] G. M. Weinberg. *An introduction to general systems thinking (silver anniversary ed.)*. Dorset House Publishing Co., Inc., New York, NY, USA, 2001.

[38] G. B. West, J. H. Brown, and B. J. Enquist. A general model for the origin of allometric scaling laws in biology. *Science*, 276(5309):122–126, April 1997.