# Using Honeynets for Internet Situational Awareness

Vinod Yegneswaran* vinod@cs.wisc.edu
Paul Barford* pb@cs.wisc.edu
Vern Paxson† vern@icir.org

*Abstract*—**Effective network security administration depends to a great extent on having accurate, concise, high-quality information about malicious activity in one's network. Honeynets can potentially provide such detailed information, but the volume and diversity of this data can prove overwhelming. In this paper we explore ways to integrate honeypot data into daily network security monitoring with a goal of sufficiently classifying and summarizing the data to provide ongoing "situational awareness." We present such a system, built using the Bro NIDS, and discuss experiences drawn from six months operation.**

**One key aspect of this environment is its ability to provide insight into large-scale events. We look at the problem of accurately classifying botnet sweeps and worm outbreaks, which turns out to be difficult to grapple with due to the high dimensionality of such incidents. Using datasets collected during a number of these events, we explore the utility of several analysis methods, finding that when used together they show promise for contributing towards effective situational awareness.**

## I. Introduction

Effective network security administration depends to a great extent on having accurate, concise, high-quality information about malicious activity in one's network. However, attaining good information has become increasingly difficult because the profile of malicious traffic evolves quickly and varies widely from network to network [7], [2], and because security analysts must discern the presence of new threats potentially hidden in an immense volume of "background radiation".

In addition, much of the information available to security analysts from sources such as intrusion detection systems comes in the form of pinpoint descriptions of low-level activities, such as "source *A* launched attack *CVE-XXX* against destination *B*". Standard best practices rarely include automatically acting on such information due to the prevalence of false and redundant alarms. In addition, the information often lacks sufficient breadth for forensic or root cause analysis.

The long-term objective of our work is to elevate the quality and timeliness of information provided to network security analysts. We appeal to the notion of network *situational awareness* as a means for defining information quality. Situational awareness is a military term referring to "the degree of consistency between one's perception of their situation and the reality" [5] or to having "an accurate set of information about one's environment scaled to specific level of interest" [6].

We envision Network Situational Awareness (NetSA) as analysts with accurate, terse summaries of attack traffic, organized to highlight the most prominent facets. NetSA should also supplement these reports with drill-down analysis to facilitate countermeasure deployment and forensic study. For example, a NetSA environment should enable an analyst to quickly assess high-level information such as the cause of an attack (*e.g.*, a new worm, a botnet, or a misconfiguration), whether the attacker specifically targeted the victim network, and if this attacker matches one seen in the past.

This work represents our initial foray into developing a NetSA environment. We pursue such an environment by coupling the use of *honeynets* for capturing large-scale malicious activity, unpolluted by benign traffic, with the application-level analysis capabilities of the Bro intrusion detection system [8]. Our approach is geared towards developing *building blocks* for a NetSA architecture that can provide SA ranging from real-time event notification to forensic analysis of large scale events. (A non-goal for this initial work is providing an "automated Big Picture" that achieves the flexibility and robustness that we ultimately envision.)

Honeypots are Internet systems deployed for the sole purpose of being compromised in order to assess adversaries. Networks of honeypots are termed *honeynets* [4] and, like network telescopes, are typically deployed on otherwise unused address space. Systems such as Honeyd, iSink and the Internet Motion Sensor simulate honeynets by using *network-level* active responders [9], [10], [1]. These systems offer the benefit of fine-grained attack analysis without the associated control issues of high-interaction honeypots, *i.e.*, no need to manage real systems and deal with them being actually compromised. The ability of honeynets to monitor large amounts of address space [10] makes them an appealing source of timely information on new outbreaks and scanning attacks.

Also related is our previous study describing the broad characteristics of Internet "background radiation" using data collected from honeynets [7]. Effectively analyzing the potentially vast quantity of data collected at these networks can prove challenging. Here, we focus on automating the process of honeynet monitoring. To do so, we use Bro to organize and condense the honeypot data into situational awareness summaries that can be quickly scanned for large-scale events. Our system presently highlights two classes of such events: *new* activity (i.e., an application-level abstraction not previously seen) and *spikes* of activity of a type previously seen, but now occurring with an unusually large number of offending sources.

The initial goal towards which we work is to accurately attribute such events as due to either $(i)$ new worm outbreaks, or $(ii)$ "botnet" [3] sweeps. We pursued this objective by developing a set of statistical analyses that consider source-arrival and scanning patterns to characterize different features of large-scale events. We report experiences from operating a prototype of our NetSA environment over the past six months, from which we have constructed a corpus of 22 large-scale events that include well-known worm outbreaks, botnet sweeps, and misconfiguration. Our work remains preliminary in that we do not yet have the capability to consistently discriminate among these types, but the combinations of our methods appear promising.

*Dept. of Computer Science, University of Wisconsin at Madison
†International Computer Science Institute; Lawrence Berkeley Laboratory

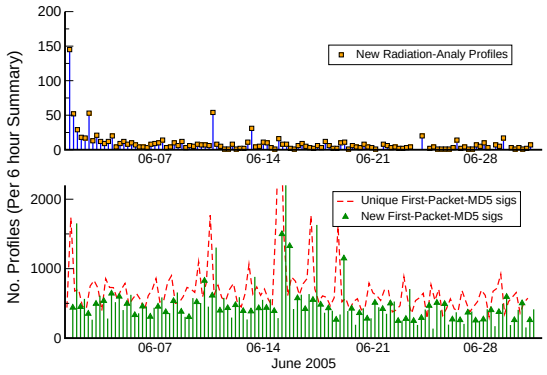


Fig. 1. Comparison of *Radiation-analy* vs. first-packet MD5 signature summarization; *Radiation-analy* provides a significant benefit.

## II. System Structure

In this section we describe the system components we built for acquiring the necessary inputs to NetSA. These include initial filtering of the raw telescope feed; engaging the sources that survive the filtering in dialog; abstracting the dialogs into their semantic elements; identifying semantic elements previously unseen; determining which recent activity merits the attention of the operator; and extracting different features of large-scale events to gain insight into their nature. Note that crucial to effectively identifying "previously unseen" semantic elements is building up a longitudinal baseline against which to compare recent observations. We discuss our experiences with operating the system over time in order to do so in the next section.

Our NetSA system includes the following components:

• **Tunnel filter** sends traffic from the monitored address space to the active responders using UDP encapsulation. The tunnel employs a simple *one-source* → *one-destination* filter: we allow each source to talk to only the first destination it contacts. This filtering greatly reduces the amount of traffic seen by responders without a substantial impact on the overall attack profiles [7]. Our analysis includes both filtered and prefiltered data.

• **Active responders** are a collection of service emulators, running in Honeyd or iSink. The responders enable fine-grained attack analysis by engaging sources in packet exchanges for specific services. We presently run responders on a number of commonly exploited services, including NetBIOS/SMB (ports 137/139/445), DCE/RPC (135/1025), HTTP (80), Mydoom (3127), Beagle (2745), Dameware (6129), MS-SQL (1433), and a generic "echo-responder" for other ports. Details on responders are provided in [7].

• **Radiation-analy** is a collection of Bro policy scripts we constructed to analyze data from active honeypots. Our objective was to enable accurate high-level classification of attack profiles. This is challenging due to the complexity of the dominant protocols such as NetBIOS and MS-SQL. In developing the scripts we aimed to strike a balance between specificity and generality so as to group together activity that is semantically equivalent from an attack perspective even if not identical as transmitted. We achieve this through two means: ($i$) considering protocol as well as "well-known" exploit semantics ($ii$) aggregating activity at multiple granularities.

The *Radiation-analy* scripts generate summaries at several granularities: ($i$) per-source scanning profiles, ($ii$) connection-level summaries to distill into aggregate source counts of identical connection profiles, and ($iii$) aggregate summaries of *session* profiles, where a session can be comprised of multiple types of connections. We use the first two in our daily evaluations; aggregating session profiles has proven more difficult due to the diversity seen across groups of connections.

In Figure 1, we compare the effectiveness of *Radiation-analy* summaries with the simple first-packet-MD5 classification strategy proposed in [1] (*i.e.*, compute an MD5 signature of the first payload seen to test if it matches a previous payload) over one month. We see that the *Radiation-analy* summaries enable the system to quickly learn about attack profiles, and after the first couple of six-hour summaries we typically see fewer than 10 new profiles per six-hour interval, a time-scale suitable for manual supervision. In contrast, the first-packet-MD5 signature caching produces hundreds of new profiles per interval and does not "learn" well .

• **Adaptation**. A key aspect of our framework is that it automatically updates its notions of types of activity over time. Specifically, when activity fails to match an already-known profile, the system inserts a description of the new activity into a MySQL database so that in the future it will be identified as something previously seen. We base these descriptions on semantic-level "tags" derived from Bro's application-level analysis. Two examples are "`445/tcp, binary-upload, CREATE_FILE: "lsarpc"`" and "`RPC bind: afa8bd80-7d8a-11c9-bef4-08002b102989 len=72; RPC request (24 bytes)`".

Our operational deployment of *Radiation-analy* produces situational summaries in 6-hour batches. These update the Honeynet database and feed *Situational-analy*, each described next.

• **Situational-analy** is a script that queries the Honeynet database and generates periodic summaries organized to highlight new events (those not previously in the MySQL database), large-scale or unusual events, and endemic activity. To identify large-scale and unusual events we compute the deviation of an event's source volume (*i.e.*, the number of distinct source IP addresses) from that seen in the past for that type of event. We compute the deviation as a ratio, denoted $\beta$, as follows. Let $p_i$ be the number of sources with connection profile $p$ in time interval $i$, and $m$ the number of intervals prior to $i$ where we previously observed this profile. Then $\beta_{p_i} = mp_i / \sum_{j=0}^{i-1} p_i$, *i.e.*, the number of sources observed in $t_i$ divided by the mean number of sources observed for this activity, ignoring intervals when it was not observed.

*Situational-analy* generates situational summaries for high-$\beta$ events. From our experiences so far, we have found that a threshold of $\beta_{p_i} > 3.0$ for minor escalations and $\beta_{p_i} > 10.0$ for high-$\beta$ events works satisfactorily, in terms of us often finding the corresponding events "somewhat" and "quite" interesting, respectively. We discuss this further in the next section.

• **Situational In-depth** is a series of statistical analyses we developed to classify large-scale events, described in § IV. These are presently off-line tools, though we aim to adapt for real-time classification in the future.

## III. Experiences

We have operated our NetSA system for six months on a 1,280-address honeynet. Its six hour summaries have alerted us to a host of potential new exploits and botnet incidents over this

period. While the details of each incident are not always compelling, the overall insight the NetSA system gives us in terms of isolating and summarizing events has been quite clear. The operation, not surprisingly, has required tuning and refinement over time; in particular, we gained experience with examining situational summary reports, we modified the format, $\beta$ thresholds, and the adaptive rule generator to better provide information at an appropriate level of fidelity for daily use. The situational summaries currently generated have four parts, as follows:

1. **New events:** The report first summarizes *new events*, *i.e.*, those not matching an existing profile as abstracted by the Bro *Radiation-analy* script. This part of the report includes the target port, source count, and the newly generated Bro tag for this event, which includes protocol and payload details. In our experience, the number of events in this category is typically less than 5. The target ports can vary widely, while the number of distinct sources identified for these events is usually only 1.
We have identified many different types of events using this portion of the summary, including a number of misconfigurations and several suspected new virus strains and polymorphisms. However, we initially expected that such previously unseen activity would very often prove highly interesting, reflecting significant new forms of malware, once we had operated long enough to fully populate our "known activity" database with the regular background radiation one sees. An important *negative result* is that this has not turned out to be the case. The difficulty is that the low levels of "new" activity we see also often include minor variations of previously seen activity. *The problem of perfectly generalizing activity to avoid flagging variants as "new" has proven quite difficult, and remains a challenge for future work.*

2. **High-$\beta$ events:** The next section summarizes high-$\beta$ events ($\beta > 10.0$). This component of the report aims to identify fast-scanning worms and large-scale botnet attacks. The report lists the *Radiation-analy* tag for each event along with hourly and 5-minute breakdowns of the number of unique sources observed, overlap in sources between successive time intervals, number of source /8-s and number of targets scanned. We see on the order of one high beta event per day, though they are sometimes quite bursty, and sometimes a single event spans multiple 6 hr reports. Using the tools presented below, our best assessment is that most of these have been either botnet scans or misconfigurations.

3. **Minor escalations:** The next section summarizes the "minor escalations" in volume ($\beta > 3.0$). The report lists the $\beta$ value, target port, source count, mean source count in the past, and *Radiation-analy* tag. While we hypothesize that slow-scanning worms exploiting known vulnerabilities might initially become visible here, no such worm outbreak took place during our study (nor did any outbreak of a fast-scanning novel worm). Typically this section includes on the order of 10 or fewer events.

4. **Top profiles**: The final section describes the top 10 activity profiles (ranked by distinct source IP count) observed in the 6 hr period. The report includes the target port, source count, and associated *Radiation-analy* tags. This section provides an ongoing sense of endemic activity. It is most frequently dominated by NetBIOS/SMB and DCE/RPC activity, but we see a significant diversity in terms of other forms of activity.

| Attribute | Misconfig | Botnet | Worm |
|---|---|---|---|
| **Source Arrivals:** | | | |
| Temporal source counts | sharp onset | gradual | sharp onset |
| Arrival window | narrow | narrow | wide |
| Interarrival distribution | exponential | exponential * | super-exp |
| **Dst/Src Net Coverage:** | | | |
| Dest-net footprint | hotspots | binomial | binomial |
| First-dest preference | hotspots | variable | binomial |
| Source-net dispersion | low-med | low-med | high |
| **Source Macro-analysis:** | | | |
| Per-source profile | hotspots | variable | variable |
| Target scope | IPv4 | <= /8 | IPv4 |
| Source lifetimes | short | short | persistent |

TABLE I SITUATIONAL AWARENESS ATTRIBUTES SUMMARY

| Incident Name | Type | Date | No. Sources |
|---|---|---|---|
| BitTorrent | NAT misconfig | 2005-01-12 | 25 (1) |
| eDonkey1 | P2P misconfig | 2005-02-02 | 389 |
| eDonkey2 | P2P misconfig | 2005-02-06 | 709 |
| eDonkey3 | P2P misconfig | 2005-02-08 | 1,034 |
| NB HiddenShare | Botnet | 2005-01-31 | 246 |
| MS-SQL1 | Botnet | 2005-01-09 | 104 |
| MS-SQL2 | Botnet | 2005-02-01 | 245 |
| MS-SQL3 | Botnet | 2005-02-03 | 176 |
| MS-SQL4 | Botnet | 2005-02-07 | 1,953 |
| NB Incomplete | Botnet | 2005-01-10 | 6,561 |
| DCERPC_p1025 | Botnet | 2005-01-10 | 775 |
| DCERPC_p135 | Botnet | 2005-04-03 | 782 |
| DCERPC_p135-2 | Botnet | 2005-01-29 | 528 |
| p6101-unknown | Botnet | 2005-01-20 | 30 |
| NB Testfile | Botnet | 2005-01-15 | 96 |
| NB Wkssvc | Botnet | 2005-01-11 | 26,010 |
| CodeRed I | Worm | 2001-07-19 | 154,666 |
| CodeRed I Re-emergence | Worm | 2001-08-01 | 126,311 |
| CodeRed II | Worm | 2001-08-04 | 114,034 |
| Nimda | Worm | 2001-08-18 | 139,351 |
| Witty | Worm | 2004-03-20 | 5,553 |
| Slammer Re-emergence | Worm | 2005-03-18 | 350 |

TABLE II SUMMARY OF HIGH-$\beta$ INCIDENTS & WORM OUTBREAKS

## IV. SITUATIONAL AWARENESS IN-DEPTH

In this section we present a set of nine statistical analyses we developed with the objective of effectively classifying large-scale events. We base each on a hypothesis about the expected behavior of three major classes of large-scale events: worm outbreaks, botnet sweeps, and misconfigurations. Table I summarizes the expected behavior for each type of event from the perspective of a honeynet. In particular, we focus on source arrivals (a probe from a distinct source IP), individual source characteristics, and network coverage, as discussed in the following sections. While we have yet to identify a single method that works in all cases, taken together these analyses provide a broad perspective on large-scale events.

Two elements of the table merit clarification, both concerning "Interarrival distribution". For this row, "exponential" indicates interarrivals consistent with a Poisson process, *i.e.*, independent arrivals that occur at a constant rate. The "exponential*" entry for Botnets indicates that initially we expected botnet probing to arrive in an *impulse*, rather than as a Poisson process; but, for reasons discussed below, the latter is often instead the case. The "super-exp" entry for Worms reflects that while at a given instant in time we might expect arrivals to appear Poisson (assuming the worm is random-scanning with a well-seeded random number generator), we also expect the worm's activity to grow over time as it spreads, so we anticipate seeing a Poisson process whose rate steadily increases until the worm attains saturation.

We collected traces for 22 large-scale events, detailed in Table II, to evaluate the utility of each type of analysis. We

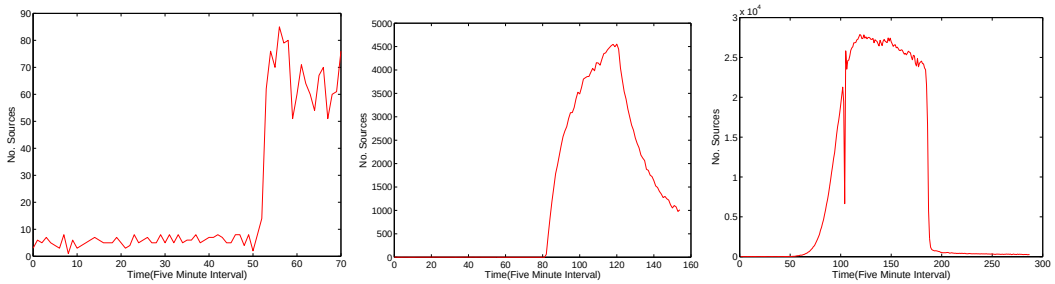


Fig. 2. Temporal Source Counts in five minute intervals for (left to right) eDonkey3, Wkssvc Botnet and Code-Red 1

collected the traces for the misconfiguration and botnet events from our honeynet deployment, while the worm outbreak traces (other than the Slammer resurgence) came from various archival sources. We now turn to a discussion of each type of analysis.

### A. Source Arrivals

• **Temporal source counts:** We hypothesized that a botnet sweep would be characterized by a sharp rise and sharp decay in temporal source counts, as the botnet was first ordered *en masse* to probe, and then completed its probing. In contrast, we expected the growth of worms to reflect the size of the infected population, so the scanning behavior would steadily increase until the worm shut down (*e.g.*, Code Red 1) or was cleaned up.

We evaluate this by considering the scanning activity in terms of the number of distinct sources seen in successive time intervals. Figure 2 shows the temporal source counts for three different events: probing of a specific honeynet address that appears due to a misconfiguration in the eDonkey peer-to-peer file sharing system (left); probing for the Windows `wkssvc` service in an event we believe is most plausibly attributed to a botnet; and historical data from the initial outbreak of Code Red 1 on July 19, 2001. (This last plot exhibits a brief measurement outage at the sharp line towards the left.) While all three events exhibit a relatively sharp onset, that for eDonkey is particularly sharp, Wkssvc is concave down, and Code Red 1 is concave up. These potentially reflect three different types of activity onset: sudden propagation among the sources (eDonkey), propagation that reaches most of the sources quickly but takes time to find all of them (`wkssvc`), and the logistic growth characteristic of a worm (Code Red 1). In addition, the probable botnet activity is distinguished from the others by its gradual but steady decay.

• **Arrival window:** We next look at the nature by which *new* sources arrive. We initially expected that botnets would exhibit a sharp spike in new arrivals as the master of the botnet pushed out probing commands to each bot. However, this turns out to often not be the case. As we confirmed by analysis of source code from a widely used botnet controller (phatbot), a common way of structuring botnets is not to push commands to them but rather to have the bots *poll and pull*. For the source code we examined, bots wake up every 1000 seconds to check for new commands. Given this behavior, rather than a sharp onset we instead might expect a steady rate of arrival over an interval of 10–20 minutes.

Figure 3 shows the arrival rate for three events. The Wkssvc event is clearly more regular than Nimda, but spread out over 10,000 sec. Perhaps this reflects a botnet with a polling interval of 10,000 sec rather than 1000 sec; but, by itself, we cannot really tell, so we find that the arrival window of new sources is insufficient by itself to distinguish worms from botnets.

• **Interarrival distribution:** If bots indeed poll independently for the instructions, then they will activate with a *uniform distribution* over the polling interval. If in addition the rate at which the bots then reach the honeynet with their probing is independent of when they receive their instructions, then we would expect the arrival of the new sources to also be uniformly distributed over the polling interval; *i.e.*, the arrivals will appear to form a Poisson process, resulting in exponentially distributed interarrival times. On the other hand, the source interarrival times from worms should exhibit an increasing rate while the worm initially propagates.

To evaluate source interarrival characteristics, we break up events into successive intervals, each with an equal number of sources (*e.g.*, we pick 10 intervals each with 10% of new sources). We then plot the distribution of interarrival times and compare against an exponential reference distribution fitted to the mean. We are unable to show graphs due to space constraints, but for an evaluation over all the events in our set, we find that botnet and misconfiguration events often show consistency with exponential interarrivals; Worm outbreaks do so, as well, but with *different rates for different intervals*.

### B. Destination/Source Net Coverage

• **Destination-net scan footprint:** Another set of salient features for large-scale events concerns which destinations they probe. We would expect misconfigurations to target only a few addresses, while botnets and worms may or may not exhibit localized scanning, which might be structured (more likely for botnets, we might think) or might be randomized (more likely for worms).

We evaluate this behavior by considering the number of scans per source and the number of sources that scanned particular destination IP addresses. Figure 4 shows the destination network scan footprint for three different events. The eDonkey misconfiguration event clearly shows hotspots, while the target selection for the worm and botnet scenarios visually appear random and comprehensive.

• **First destination preference** Next, we test for a preference in the first destination chosen by sources. This can reveal trends such as botnet sources that always start sequentially scanning from the top of subnet, or sources of bias in the random number generators used by worms to select targets. To evaluate this behavior, we count the number of times each destination address was chosen by a source as its first target. If the scanning process is entirely random, *i.e.*, there is no bias in the scanning order,

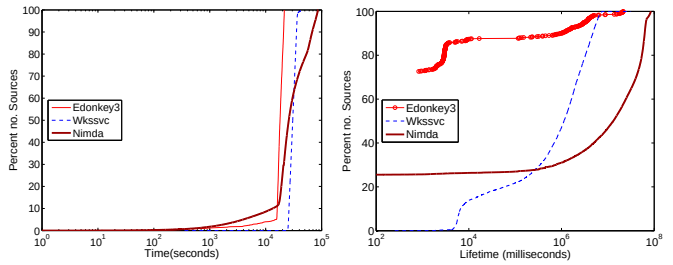


Fig. 3. Left: CDF of source arrivals eDonkey3, wkssvc, Nimda; Right: CDF of source lifetimes

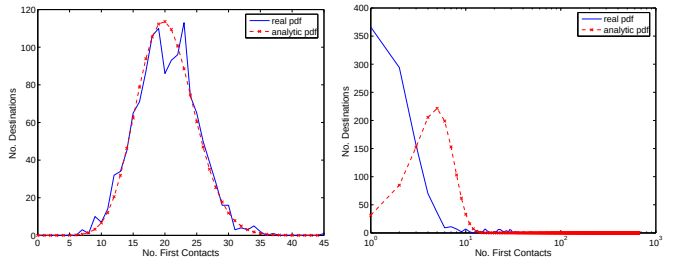


Fig. 5. PDF of first destination preference (left) Wkssvc Botnet Incident: 2005-01-11 (right) Nimda: 2001-09-18

then we would expect these counts to have a binomial distribution in terms of $n$ trials ($n$ = # sources) and a probability of success (*i.e.*, a given destination is visited first) $p = 1/1280$ (size of 5 /24 networks monitored).

Figure 5 plots the first-destination preference for two events, along with the expected values from the corresponding binomial distribution. The Wkssvc botnet fits the binomial quite well, indicating it chooses its destinations fully at random, while Nimda exhibits a local preference. Not surprisingly, an eDonkey misconfiguration event (not shown) shows a complete lack of fit.

• **Source-net dispersion:** Next, we consider the distribution of source hosts across the IPv4 address space. We hypothesize that hosts observed in worm outbreaks will be much more broadly spread across the address space than botnets. Since sources sending traffic to the honeynet interact with an active responder (other than for single-packet UDP probes), we can generally eliminate the possibility of spoofed source IPs. We then compute a histogram of the count of sources seen from each /8 address aggregate. Such plots (not shown) reveal that the source dispersion of known worm outbreaks is much higher than that for likely botnet sweeps or misconfigurations.

### C. Source Macro-analysis

• **Per-source scanning profile:** Next, we investigate the degree to which the scanning profiles of individual sources can provide insight into a large-scale event's aggregate behavior. To do so, we randomly select up to 100 sources and plot the destinations that each visit, sorting on the lowest destination address visited (an alternative might be to sort by arrival time). In addition, we construct phase-space plots of the consecutive honeynet addresses a given source probes. These two plots are complementary: one provides per-source *coverage* information, the other provides per-source *ordering* information.

Figure 6 shows examples of scan profiles and associated phase-space plots for an MS-SQL event likely due to a botnet. The left plot reveals two types of sources, those covering the

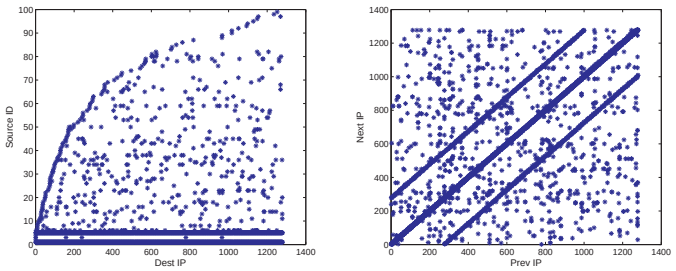


Fig. 6. Left: Dest Scanning profiles of 100 random sources ordered by first destination. Right: Phase plot of successive destination IPs scanned in local network the MSSQL Botnet Incident: 2005-02-03

entire target space and those scanning a small number of IPs. The phase-space plot on the right suggests that the former set of sources scan the address space sequentially. We also see two parallel lines on either side instead of a single diagonal line. This artifact is consistent with a single source using *two independent scanning threads*, each of which traverses the address space separately but at the same rate. We see similar plots for other botnet incidents, suggesting that we need to account for such concurrent scanning when testing for sequential scanners.

• **Inferring target scope:** A general situational awareness question concerns how broadly a given event was *actually* scoped, as opposed to its prevalence seen within the honeynet. It can matter a great deal whether a given event specifically targeted the monitored network, or only incidentally probed it as part of much broader probing activity.

Roughly, we would expect worms to tend to have global target scope, with botnets and misconfigurations considerably more localized. The problem then becomes how to assess global scope given only a single honeynet vantage point. To do so we try to infer and then compare the global scanning rate of each source versus its local (within the honeynet) scanning rate.

We base our method for doing this on the observation that retransmitted TCP SYN packets will generally be sent within 3 seconds. We can often estimate how many packets a source has sent between two observed packets by changes in the IP ID counter (if the source implements the common policy of incrementing the ID by one for each packet sent). A 3-second interval is sufficiently short such that it is highly unlikely the IP ID field will have fully wrapped (*i.e.*, the source sends $>$ 65,535 packets). Thus, the IP ID spacing between retransmitted SYN packets gives an estimate of the source's global scan rate. We can extend this trick (which admittedly will often not work for sources that craft their own packets) to UDP sources by considering packets we observed that arrive $\leq 3$ sec apart.

In addition, we can construct an estimate of the *local scanning rate* for each source by dividing the number of probes from it by its lifetime. We can then estimate the *broader reach* of a source as the ratio of its estimated global rate versus its estimated local rate, multiplied by the size (in addresses) of the honeynet.

Figure 7 shows log-log scatter plots of the estimated global and local scanning rates of each source during four different events, confining the plot to sources that contacted at least 5 destinations. For the eDonkey misconfiguration event, we estimate a multiplier between $10^3$ and $10^5$, so we infer that the misconfiguration does not simply target our honeynet but includes thousands of other targets. The `Wkksvc` botnet incident yields an es-

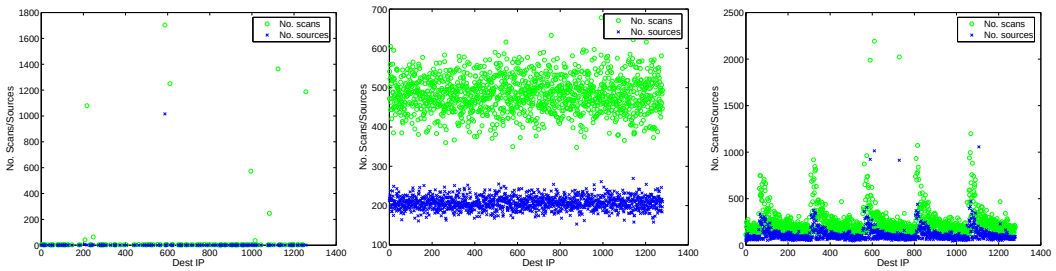


Fig. 4. Destination Net Scan Footprint (left) eDonkey Misconfiguration: 2005-02-08 (middle) Wkssvc Botnet Incident: 2005-01-11 (right)Nimda: 2001-09-18

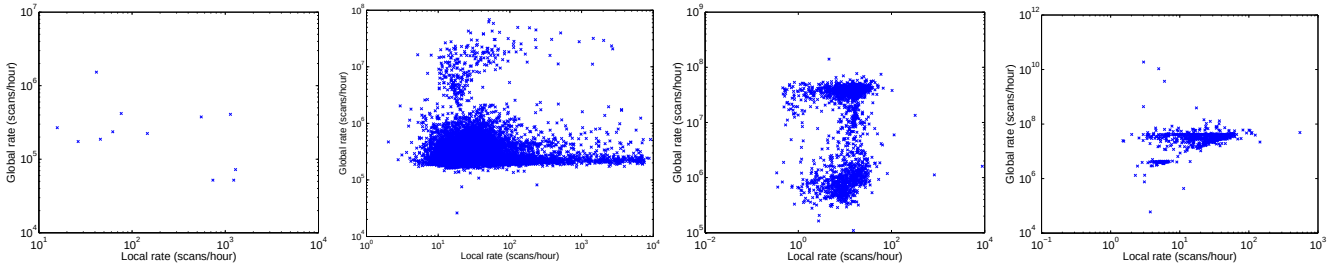


Fig. 7. Log-Log plot of global and local scanning rate, ratio provides an estimate of the instantaneous global footprint (left) eDonkey Misconfiguration (2005-02-08) (second) Wkssvc Botnet Incident 2005-01-11 (third) Nimda Worm Outbreak: 2001-09-18 (right) Witty Worm Outbreak [UW Net]: 2004-03-20

timated multiplier of around $10^4$, indicates that the event likely targeted the equivalent of a /8 network. This level of scoping holds for all of the botnet incidents we have analyzed. On the other hand, data from Nimda reveals two clusters of sources. The multiplier here for the higher cluster is between $10^6$ and $10^7$, consistent with the entire IPv4 address space. Finally, data from the Witty worm outbreak yields an estimated multiplier of around $10^6$. The target network used to collect that data had $\approx$ 8K addresses, so this scales up to a footprint on the order of the entire IPv4 address space, the correct scope for Witty.

• **Source lifetimes:** The final attribute we consider is the lifetime of sources, *i.e.*, for how long do we see them active in the honeynet. We hypothesize that botnet sources will be short lived, since they presumably are told to conduct a specific scan and will stop when they have completed it, while worm sources will be persistent unless they have mechanisms in them to stop scanning after a certain point, which have not been seen to date (other than Code Red 1's die-off on the 20th of each month).

Figure 3 plots the CDF of source lifetimes for three events. A lifetime of 0 corresponds to seeing the source very briefly or perhaps only once. We see that our expectation largely holds: botnets and misconfigurations have short lifetimes, while the lifetimes of worm sources are distributed broadly. While this analysis can be a useful discriminator between worms and botnets, its utility is limited by the fact that we need to wait before we can make the determination.

## V. Conclusions and Future Work

Our quest in this study is to enrich the set of information at a security analyst's disposal by creating Internet Situational Awareness. We base our study on the premise that honeypot data can provide a source of timely, accurate and concise information for situational awareness—but this data must be organized and condensed to be useful. To that end, we developed a system based on honeynets, analyzers that leverage the Bro NIDS, and a MySQL backend database to facilitate analysis of honeynet data. This system has captured and identified numerous interesting events during our six-month preliminary deployment.

An important component in our NetSA environment is the statistical analyses we developed to gain insight into large-scale events. While we have not yet attained an integrated analysis regimen that accurately and automatically classifies these events, the individual analyses each provide useful perspectives. As this study continues, we plan to explore real-time classification as well as working towards such an integrated analysis.

**Acknowledgements:** Ruoming Pang did the initial design of *Radiation-analy* during our joint work in [7]. We thank Cristi Estan and the anonymous referees for helpful comments on earlier versions of this paper. This work is supported in part by NSF grants CNS-0347252, ANI-0335234, CCR-0325653, NSF-0433702, and STI-0334088. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

## References

[1] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson. The Internet Motion Sensor: A Distributed Blackhole Monitoring System. In *Network and Distributed Security Symposium*, San Diego, CA, January 2005.

[2] E. Cooke, M. Bailey, M. Mao, D. Watson, F. Jahanian, and D. McPherson. Toward understanding distributed blackhole placement. In *Proceedings of CCS Workshop on Rapid Malcode (WORM '04)*, October 2004.

[3] German Honeynet Project. Tracking Botnets. http://]www.honeynet.org/papers/bots, 2005.

[4] The Honeynet Project. http://project.honeynet.org, 2003.

[5] Navy Aviation Schools Command. Situational Awareness. https://www.cnet.navy.mil.crm/crm/stand_mat/seven_skills/SA.asp, 2005.

[6] Network Centric Operations Industry Consortium. Situational Awareness. http://www.ncoic.org/download/NCOIC_Lexicon_v8.pdf, 2005.

[7] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet Background Radiation. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, 2004.

[8] V. Paxson. BRO: A system for detecting network intruders in real time. In $7^{th}$ *USENIX Security Symposium*, San Antonio, Texas, January 1998.

[9] N. Provos. A virtual honeypot framework. In *Proceedings of USENIX Security Symposium*, San Diego, CA, August 2004.

[10] V. Yegneswaran, P. Barford, and D. Plonka. On the design and use of Internet sinks for network abuse monitoring. In *Proc. RAID*, 2004.