The Dark Oracle: Perspective-Aware Unused and Unreachable Address Discovery

Evan Cooke, Michael Bailey, Farnam Jahanian
Electrical Engineering and Computer Science Department
University of Michigan
{*emcooke, mibailey, farnam*}*@umich.edu*

Richard Mortier
Microsoft Research
Cambridge, UK
mort@microsoft.com

Abstract

Internet traffic destined for unused or unreachable addresses provides critically important information on malicious and misconfigured activity. Since Internet address allocation and policy information is distributed across many devices, applications, and administrative domains, constructing a comprehensive map of unused and unreachable ("dark") addresses is challenging. In this paper, we present an architecture that automates the process of discovering these dark addresses by actively participating with allocation, routing, and policy systems. Our approach is to adopt a local perspective revealing unreachable external addresses and unused private and local addresses, and enabling the detection of threats coming into and out of a network. To validate the approach, we construct a prototype system called the Dark Oracle that uses internal and external routing data and host configuration information, such as DHCP logs, to automatically discover dark addresses. We experimentally evaluate the prototype using data from a large enterprise network, and a regional ISP, and from deployment of the Dark Oracle on a large academic network.

1 Introduction

It was once widely believed that the Internet was in imminent danger of address exhaustion due to millions of new users and the proliferation of new devices. Instead, we now find huge numbers of unused addresses. Large address blocks are still not allocated by registries, blocks allocated to organizations are never externally advertised or routed, and there are millions of unused addresses within allocated and routed subnets between the laptops, desktops, and servers we use every day. During the course of this study we found that 66.8% of all possible IPv4 addresses were never announced through BGP, 57.5% of the addresses assigned to the campus of a large academic network were never internally routed, and 64.8% of the addresses allocated to a DHCP server were never assigned to a host.

This vast pool of unallocated, unrouted, and unassigned addresses sitting idle across the Internet can be used to provide intelligence on malicious and misconfigured Internet activity [24]. There are a range of techniques for monitoring contiguous ranges of unused addresses, including honeypots [1, 30, 31], virtual honeypots [3,15,35], emulators [26,37], simple responders [2], and passive packet capture [11, 28]. We refer to these techniques together as *honeynet* monitoring.

Existing honeynet monitoring systems only cover a very small percentage of the available unused address space. Two fundamental problems limit monitoring more addresses. First, address allocation information is distributed across many devices, applications, and administrative domains. For example, address registries like ARIN can provide information on what addresses are assigned to an organization, but not on what addresses are routed or reachable. The second challenge is that address allocations can change quickly. For example, wireless devices can enter and leave a network, and instability in routing information can impact address reachability. The result is that honeynet monitoring systems today monitor only easily obtainable, contiguous blocks of addresses.

This paper presents an architecture that automates the process of discovering these non-productive addresses by participating directly with allocation, routing, and policy systems. The goal is to pervasively discover unused and unreachable ("dark") addresses inside a network so that traffic sent to those addresses can be forwarded to honeynet monitoring systems.

This architecture is fundamentally different from existing systems because it is *perspective-aware*. This means it adopts the local perspective of a specific network, thereby expanding the number of monitorable addresses and enabling *outgoing* honeypots. Today, threats coming into a network [32, 33] receive the most attention; however, threats inside the network, such as infected laptops, are arguably more serious. The proposed architecture discovers addresses that are externally *unreachable* from the perspective of a particular network, and it routes any packets leaving the network that are destined for unreachable addresses to a honeynet. These outgoing monitors provide unique visibility into local infections and misconfigurations.

To demonstrate our approach, we construct the *Dark Oracle*, a system designed to discover unused and unreachable addresses within a network. The system integrates external routing data like BGP, internal routing data like OSPF, and host configuration data like DHCP server logs to construct a locally accurate map of dark addresses. The Dark Oracle automates address discovery, significantly simplifying the process of finding dark addresses. It also provides unique local visibility into internal threats and targeted attacks.

We experimentally evaluate our approach using data from a large enterprise network, and a regional ISP, and from deployment of the Dark Oracle on a large academic network with more than 10,000 hosts. We show how the external, internal, and host configuration address allocation data sources are stable over time, and that the system is scalable. Even when each data source is sampled just once a day, the error in address classification is well under 1%. We deploy a pervasive honeynet detector that uses the addresses from the Dark Oracle and show how unused addresses from a DHCP server reveal almost 80,000 unique source addresses compared to 4,000 found by a traditional /24 monitor. Because we are also able to monitor outgoing addresses, we discover almost 2,000 locally infected or misconfigured hosts in an academic network. These experiments demonstrate the effectiveness of the Dark Oracle in discovering highly distributed local and global dark addresses, thereby enabling quick detection of targeted and internal attacks.

2 Background and Related Work

As Internet-based attacks have become increasingly commonplace and complex, it has become impractical for experts to manually analyze each attack and the hundreds of subsequent variants [9]. This rapid growth in malicious Internet activity has driven the need for more automated data collection and analysis systems.

Approaches to the detection and characterization of network-based threats fall into two general categories: monitoring production systems such as live networks or host-based firewalls [33], and monitoring non-productive *honeypot* resources. This paper focuses on honeypots which provide a unique pre-filtered source of intelligence on the activity of attackers and other anomalous processes [6, 30].

Host-based honeypot systems have traditionally been allocated a single IP address which limits visibility into processes such as random scanning threats [30]. This limitation of monitoring only a single address helped to motivate the development of wide-address monitors called network telescopes [21], sinks [37], blackholes [29], and darknets [11]. These efforts have produced a new understanding of denial of service [22], worms [2, 4, 20, 28], and malicious behavior [24].

Monitoring large numbers of unused addresses simultaneously has been shown to provide quicker and more complete information on threats [8, 16, 17, 21]. Cooke *et al.* demonstrated that distinct honeynets observed orders-of-magnitude different amounts of traffic and different numbers of unique source IPs [8, 10]. These differences persisted even when accounting for local preference and specific propagation algorithms. Pang *et al.* also demonstrated that data collected at honeynets at three locations belonging to three distinct networks differed significantly [24]. Kumar *et al.* recently demonstrated how the Witty worm's random number generator produces non-uniform scanning [17]. However, gathering the same detailed forensic information produced by a real honeypot is a scalability challenge. One approach is to trade fidelity for scalability by emulating operating systems and services rather than running real operating system or application instances [26, 37].

Another approach is to place each honeypot instance within a virtual machine [15, 32]. This enables the execution of multiple operating systems on a single physical machine. Unmodified virtual machines are not sufficiently scalable because a large monitor can receive hundreds or thousands of connections per second. One way of reducing this load is to filter the incoming connections before they reach a honeypot [3]. Another technique is to make the process of storing and spawning virtual machines more efficient. The Potemkin Virtual Honeyfarm [35] uses *copy-on-write* virtual machine images to quickly restore and execute operating system images as packets enter the honeyfarm.

In summary, techniques that monitor unused addresses provide important intelligence on new Internet threats and are becoming more operationally important as Internet-based attacks have become both increasingly commonplace and complex. Recent honeynet scalability advances have provided the framework for monitoring larger and more diverse address ranges and in this paper we attempt to address this need by developing a system designed to pervasively discover these addresses.

3 Redefining Dark Space

When most researchers refer to honeypots, honeynets, darknets, network telescopes, and blackholes there is an implicit assumption that the monitored addresses are globally advertised and globally reachable. That is, a path that exists from most points on the global Internet to the monitored addresses.

This view deserves closer scrutiny. We propose that the number of possible dark addresses would greatly increase if the definition is expanded to include *unreachable* addresses. By adopting the perspective of a particular network, it is possible to discover addresses that may or may not be reachable in other parts of the Internet. A

Botnet Command	Targ.	Botnet Command	Targ.	Botnet Command	Targ.
ipscan r.r.r.r dcom2 -s	*No*	ipscan i.i.i.i dcom2 -s	*No*	advscan wkssvcENG 100 0 0	*No*
adv.start lsass 198 5 0 -b	*No*	ipscan s.s.s.s dcom2 -s	*No*	ipscan r.r.r.r dcom2 -s	*No*
ipscan 24.s.s.s dcom	*Yes*	advscan dcass 300 5 0 141.x.x.x	*Yes*	advscan lsass 100 5 999 -b	*No*
advscan dcass 300 5 0 140.x.x.x	*Yes*	advscan dcass 300 5 0 140.142.x.x	*Yes*	ipscan 69.27.s.s dcom2 -s	*Yes*
ipscan 207.s.s.s dcom2 -s	*Yes*	ipscan s.s mssql2000 -s	*Yes*	ipscan s.s.s lsass -s	*Yes*
ipscan 84.9.s.s dcom2 -s	*Yes*	ipscan s.s webdav3 -s	*Yes*	ipscan r.r.r.r dcom2 -s	*No*
ipscan s.s.s mssql2000 -s	*Yes*	ipscan 194.s.s.s dcom2 -s	*Yes*	ipscan 194.116.s.s dcom2	*Yes*
advscan lsass_139 50 10 0 128.218.x.x	*Yes*	ipscan 192.s.s.s dcom2 -s	*Yes*	ipscan 128.s.s.s dcom2 -s	*Yes*

Table 1: Botnet scan commands captured on a live /15 academic network during May 2005. The table shows that 70% of the captured commands were targeted at a specific /8 or /16 network.

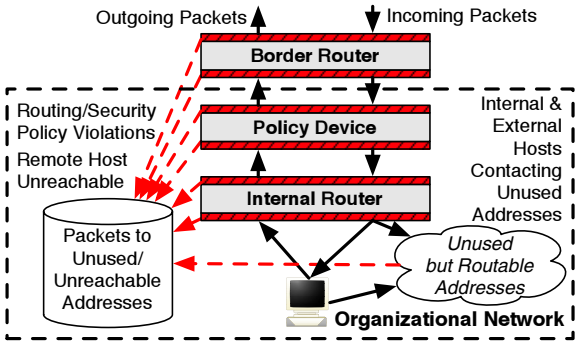


Figure 1: Unused and unreachable addresses inside a network. These addresses can come from a range of sources including routing and policy enforcement devices.

packet leaving a network that would be dropped by an upstream router because the destination address is not allocated is an operationally interesting packet and warrants closer inspection. By locating these upstream and locally unreachable addresses and combining them with unused addresses from throughout the network [13] it is possible to significantly increase the number of dark addresses available to honeynets. Some examples of dark addresses include:

Unused Addresses:

- Unused addresses that are globally advertised and *routable*
- Unused private addresses that are locally *routable*
- Unused UDP/TCP ports on an end-system

Unreachable Addresses:

- Reserved addresses
- Allocated but unadvertised addresses
- Private addresses that are locally *unroutable*
- Unused addresses that are globally advertised but *unroutable* (e.g., due to policy)

A pictorial representation of the possible sources of dark addresses is illustrated in Figure 1. Devices and configuration from the routing infrastructure and from policy enforcement mechanisms (e.g., network firewalls) are possible sources of address information. The key idea is that by using a *perpective-aware* address discovery mechanism, it is possible to find and utilize a far greater range of dark addresses.

This broader view of dark addresses provides three fundamental improvements to honeynet systems. First, highly distributed dark addresses enable the detection of targeted attacks and are more difficult to fingerprint. Second, local addresses such as unused private addresses provide a unique perspective into internal threats. Finally, a large number of addresses provides quick detection of randomly propagating threats.

3.1 Perspective-Aware

The expanded definition of dark addresses has implications on how dark addresses are monitored. It is now possible to monitor both *incoming* and *outgoing* traffic. That is, if an address is not internally or externally reachable, that address can be marked as dark. By tracking incoming and outgoing packets, one also gains a unique perspective into local behavior. Below is a list of interesting features one can detect by monitoring incoming and outgoing traffic to dark space.

- **Inbound Traffic:** Globally-scoped attacks (worms), externally-sourced targeted attacks (botnet scans), backscatter (DOS attacks), and externally-sourced reconnaissance (scans).
- **Outbound Traffic:** Locally infected machines (worms/botnets), local misconfiguration (misconfigured DNS), and internal reconnaissance (scans).

To explore the importance of having visibility into both incoming and outgoing traffic, we studied the targeting behavior of bot infected computers. Bots have the ability to perform targeted attacks against external hosts and local attacks against internal systems [9]. To investigate the prevalence of targeted bot behavior, we conducted a study of botnet commands. We looked for the specific command signatures of Agobot/Phatbot [7], rBot/SDBot [19], and Ghost-Bot in the payloads of traffic captured in a large academic network. Table 1 shows a list of commands from approximately 11 bots detected by the system during May 2005. Each command instructs the bot to begin scanning a range of IP addresses. We found that 70% of the commands were targeted at external /8 or /16 networks or specified a scan of local systems (e.g., `ipscan s.s webdav3 -s`). The implica-

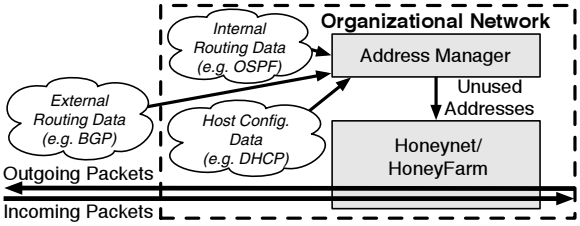


Figure 2: Major components of an automated dark address discovery architecture. Multiple sources of allocation data are used to find *unused* and *unreachable* addresses.

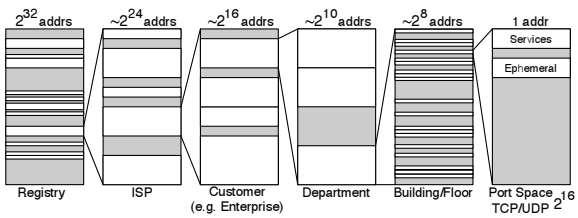


Figure 3: Example of the IPv4/port address allocation hierarchy. The allocation number above each step reflects the relative quantity of addresses being managed in the allocation process.

tion is that monitoring targeted attacks is becoming more important, and that distributed, locally-scoped monitoring is critical for obtaining a complete picture of targeted external attacks and internal threats.

In summary, a simple way to dramatically expand the visibility of honeynet systems is to monitor *unreachable* and *unused* addresses. We now describe a system designed to automatically discover these dark addresses inside a network.

4 Architecture

In this section we describe an architecture that automates the process of discovering dark addresses by participating directly with allocation, routing, and policy systems. The architecture is composed of two major components. The first component is the address allocation data sources. There are three main sources of allocation data: external routing data, internal routing data, and host configuration data. The second major architectural component is the address manager that utilizes the address allocation data to provide a map of dark addresses. A high-level diagram that depicts the major components of the architecture is illustrated in Figure 2.

In the next three subsections, we describe possible data sources for the address manager and the importance of using internal data sources. Using this understanding, we develop three classes of allocation data sources that are used as input for the address discovery architecture. Finally, we describe how to combine data from different data sources in a coherent manner.

4.1 Potential Address Sources

Discovering dark address space is challenging because address allocation information is distributed across many devices, applications, and administrative domains. This means that there is no single Internet-wide repository of fine-grained address allocation data. The situation is not much better within organizations as operators rarely have accurate per-device address allocation records. Thus, the key to obtaining accurate information is to integrate data from many sources of address allocation information. To understand where to locate this information, we first need to understand the address allocation process.

To preserve global uniqueness, IP addresses are distributed through a central authority called Internet Assigned Numbers Authority (IANA) [14]. IANA allocates large blocks of address space to regional registries such as ARIN (for North America) that handle address allocations for specific organizations. Certain organizations such as governments and large enterprises also have direct allocations from IANA. Organizations such as ISPs can then turn around and reassign regions of their allocated address space to their customers. For example, an ISP might reassign one or more sub-blocks of addresses to another smaller ISP or enterprise customer.

The addresses used within an organization are often then subdivided by campus, functional unit, or department. A DHCP server is then often used to dynamically allocate addresses to end-hosts. For example, the main site of a large enterprise network might be assigned a /16 and a specific floor within a department might have a DHCP server with the assignment of a /24 address block.

The port allocation process for end-hosts can also be considered part of the address allocation hierarchy. Unlike IP addresses, ports only need to be unique at the host-level so they can be allocated by a host without concern for global uniqueness.

A example of the allocation process is illustrated in Figure 3. The figure also shows the approximate number of addresses being managed at each step. At each step in the allocation process an organization is responsible for uniquely assigning addresses to the next step. For example, an ISP has the responsibility not to assign the same addresses to different customers. Each organization in the process must also only use or advertise addresses assigned to them. The distributed nature of the allocation process means enforcement is a challenge and there are sometimes violations that impact reachablity [18].

Figure 3 also illustrates another important concept. *At each step in the allocation process there is often a significant number of unused addresses.*

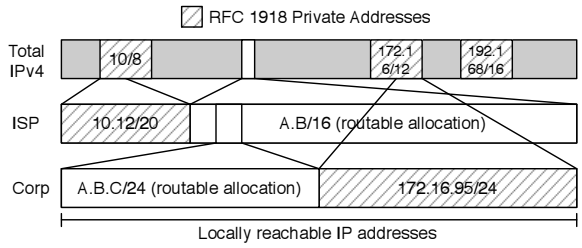


Figure 4: Usage of private address space at different levels in the address allocation hierarchy. Although not illustrated, the private address blocks used at different levels could be from the same address space.

4.2 Leveraging Internal Data

Thus far we have discussed globally unique address allocation which is only part of the process. There are also two other very important classes of dark addresses: private addresses, and policy violations. These addresses provide unique insight into local events. For example, an infected laptop configured with a 192.168.0.0/16 address from a home router is plugged into the network and immediately starts scanning. By monitoring unused portions of private address space, this type of misconfiguration and infection can be quickly identified [10].

Many organizations make extensive use of private address space. An example of private address space usage within an ISP and its customer enterprise is shown in Figure 4. It is difficult to determine the private addresses used within an organization from external data alone. Instead, by using internal routing and host configuration data the unused portions of private address space can be identified. Only small portions of private address space are typically used (10.0.0.0/8 contains 16 million addresses) so visibility into private address space can provide a large number of monitorable local addresses.

Another challenge is that both incoming and outgoing traffic can be blocked by policy applied at different levels in the address allocation hierarchy. Organizations will often use policy to strictly filter incoming traffic or to drop outgoing traffic to certain common ports. For example, an enterprise might block all outgoing TCP port 135 connections to limit outgoing file sharing. If these blocked IP/port pairs can be discovered by communicating with policy systems, then packets to those addresses can instead classified as dark.

Both unused private addresses and policy violations provide a unique source of addresses that are not typically monitored by honeynet systems. The proposed architecture supports the discovery and integration of both of these types of addresses.

4.3 Provisioning the Address Manager

The next step is to determine what data sources should be incorporated into the architecture to provide the broadest possible visibility. As we have argued, the key to discovering the broadest possible range of dark addresses is to take a local perspective. So the question is: What are the data sources available to a particular organization? We argue that there are three broad classes of address allocation data: external routing data, internal routing data, and host configuration data (as illustrated in Figure 2).

External routing data provides information on addresses that have been allocated and are routable. We use routing data rather than data from registries because registry data shows allocation which do not necessarily reflect what address are actually routable. An example source of external routing data is BGP announcements.

Internal routing data is crucial for distributing reachability information inside medium and larger organizations and provides fine-grained information on what addresses are actually allocated within an organization. For example, an ISP may advertise a full /16 through BGP but only half of that space is allocated and used internally for customers. OSPF, ISIS, and RIP are all excellent sources of internal routing data.

Host configuration data includes information from systems that allocate individual addresses to end-hosts. This includes information about address usage like unused ports directly from end-hosts, and configuration from policy devices like firewalls. Host configuration information is available from DHCP and LDAP servers which provide details on specific IP address allocations.

4.4 Synthesizing Allocation Data

Once the external routing data, internal routing data, and host configuration information reaches the address manager – as illustrated in Figure 2 – it must be synthesized into a consistent map of dark addresses. One challenge is how to resolve a conflict when two data sources disagree on the status of an address. For example, external routing data might indicate that an address was reachable while internal routing data reveals it was unused. The solution is to assign priority to the more specific data source. More specific data sources are further down in the allocation hierarchy. For example, host configuration data takes priority over external routing data.

As allocation data from many sources is brought together, it is possible to identify inconsistencies. It is expected that an address that was classified as used by a data source at the top of hierarchy might then be identified as dark by a data source at the bottom. However, if the opposite classification occurs, it can indicate a misconfiguration. For example, if a DHCP server is allocated non-private address block that is not advertised

through BGP, this can indicate that either the server was assigned the wrong addresses or there is a BGP configuration problem.

5 Dark Oracle Design/Implementation

In this section we describe the *Dark Oracle*, the realization of the dark address discovery architecture. The Dark Oracle was implemented in C and Python using a plug-in system for different address allocation data sources. The system synthesizes a list of unused addresses based on the address allocation inputs and passes that list of dark addresses to a honeynet.

In the next three subsections, we describe how we constructed the Dark Oracle using BGP external routing advertisements, OSPF internal routing advertisements, and DHCP host configuration data. We then discuss how the addresses from different data sources are combined and how we implemented a prototype honeynet using a promiscuous mode packet sniffer and a high-volume router. Finally, we discuss the issue of misclassified addresses.

5.1 External Routing Data

The source external routing data is BGP, which is the dominant exterior gateway protocol on the Internet today. The Dark Oracle obtains an up-to-date view of global BGP announcements using a feed of data from the RouteViews project [34]. RouteViews includes BGP data observed from many vantage points, so it provides a more global view of reachability than a single BGP listener in one network. Depending on the organization and upstream routing policies, it may be important to have more locally-accurate, external reachability information. In this case, it is simple to redirect the BGP module in the Dark Oracle to a local BGP feed.

To determine whether a given IPv4 address is dark, we simply check if there is a valid BGP advertisement for that address. If not, the address is declared dark. Misconfigured BGP advertisements are common across the Internet, so we first filter the advertisements using the bogon list [12].

5.2 Internal Routing Data

To capture internal routing data, the Dark Oracle uses an OSPF listener that participates in the local OSPF backbone and collects update messages [23]. In certain networks, information like router configuration could be helpful to discover details such as static routes, multiple OSPF instances, multiple areas, or other internal routing protocols like RIP. However, this information is not required by the Dark Oracle, it simply improves visibility.

To determine if given address is dark the Dark Oracle must assume the specific perspective of a particular organization. The appropriate address allocation registry, such as ARIN, is checked to decide whether an address is within the range managed by the organization and thus managed by OSPF. If the address falls within the address blocks assigned to the organization, then the current valid OSPF LSA updates are checked to see if the address is advertised. Thus, if an address is allocated to the organization and it is not advertised through OSPF, the address is classified as dark.

There are obvious complications. For example, private address space is potentially valid within an organization, so if a private address is not advertised through OSPF, it is classified as dark. It is also possible that the allocations managed by OSPF are not also assigned through the regional registry. In this case, the Dark Oracle has configuration parameters for managed address ranges.

5.3 Host Configuration Data

The host configuration data source used in the Dark Oracle uses address allocation records from a DHCP server. Rather than modify DHCP server code, the Dark Oracle can passively monitor DHCP commands on the network or directly monitor DHCP logs.

To decide whether a given address is dark, we first need to know if the address falls within the range managed by the DHCP server. To make this decision the DHCP module in the Dark Oracle requires the configured lease time and the pool of addresses from which the DHCP server allocates leases. These parameters are easily extracted from the configuration file or database and can be kept up-to-date with periodic updates. If the address is found to be managed by the DHCP server, we test to see if the address has been allocated by tracking the DHCP discover, lease, and renew messages. If the address has not been allocated, it is declared dark.

5.4 Prioritizing Data Sources

As we outlined in the previous section, the key to combining address allocation data from different sources is to assign priorities. DHCP data has the highest priority, followed by OSPF data and then BGP data. Thus, if DHCP declares an address dark, that assignment takes priority over OSPF or BGP announcements. This process is simple and easily handles additional data sources with different priority levels.

5.5 Prototype Honeynet

Once an address has been classified as dark by the Dark Oracle, that address can then be used for a range of different honeypot applications. One could use a SYN-ACK responder to elicit TCP payloads [2], a system such as *honeyd* to emulate end-host behavior [26], or even forward packets back to a honeyfarm to be executed on real end-hosts [35].

To validate the Dark Oracle we passively captured traffic to the addresses classified as dark. Passive capture is simple, scalable, and provides a large amount of information on malicious activity and misconfiguration [24]. One key piece of information provided by passively captured darknet traffic is the source IP address. The source IP address provides a good estimation of *who* is malicious and misconfigured and doesn't require any honeypot response.

We used two methods for passively capturing traffic: a program called *darktrap*, and a *blackhole* route.

5.5.1 Darktrap

The goal of *darktrap* is to process data from a promiscuous mode interface connected to a span port on a router. A span port mirrors traffic on some or all interfaces of a router to another port. Because this includes live production traffic we also constructed a mechanism to isolate packets to dark addresses.

To deal with large traffic loads (100 to 600 Mb/s), *darktrap* requires a high-speed evaluation mechanism to indicate whether a given input address is a member of a set of dark addresses. The number of addresses in the dark address pool can also be very large. For example, the BGP table can include almost 200,000 entries.

To obtain the necessary scalability we implemented a hybrid suffix-Patricia tree. Unlike a router which must find a longest-prefix match, *darktrap* requires a simpler yes/no answer if a prefix exists that covers a given address. The program uses uses a 4-level deep tree for storage in which each tree node is a 256 element-wide array. The tree is populated with the dark prefixes such that each array element in a node is set to either *NULL* (meaning no match), *-1* (meaning a /32 match), or a pointer (meaning a pointer to next level of the tree). *darktrap* was designed to be integrated into the FreeBSD kernel, but the performance was acceptable in userland. It incurred a few percent CPU overhead on a 3GHz test system, with over 600Mb/s of input traffic using the full set of prefixes from a BGP table dump on September 20, 2005.

5.5.2 Blackhole Routing

The second method we use to capture traffic to dark addresses is a *blackhole* or *fall-through* route. The idea is illustrated in Figure 5. In the example, a network is allocated 1.2.0.0/16 by the RIR, but only advertises 1.2.3.0/24 and 1.2.67.0/24 internally. Thus, the installed blackhole route, 1.2.0.0/16, captures all traffic destined for the network's allocated-but-unrouteable addresses. The idea is similar to adding a route to prevent flooding attacks against persistent loops [36]. The static route identifies all traffic to unused addresses as packets to those addresses fall-through the more specific prefixes

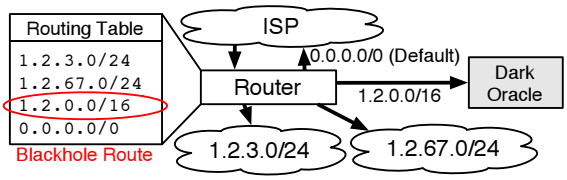


Figure 5: A blackhole route is used to capture traffic that is destined for addresses in the local network that are *not* advertised by any more specific prefix. Traffic destined for external addresses can still be successfully routed by the default route as before.

allocated to live subnets. To collect the traffic, we just placed a monitoring system next to the upstream router and configured the static route to point at the monitoring system.

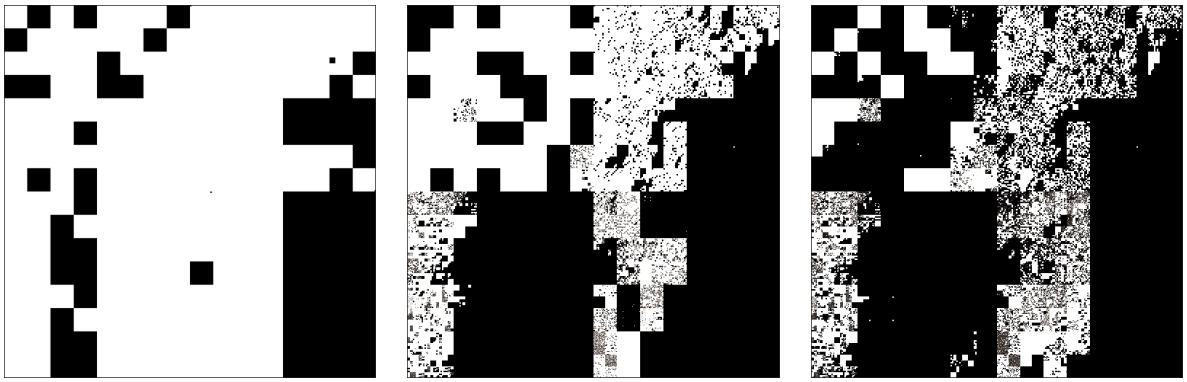
5.6 Misclassified Addresses

One important problem is misclassified addresses. That is, what if the Dark Oracle misclassifies an address as dark that should be active. There are two main reasons why an address might be misclassified: (1) the state between the Dark Oracle and a data source becomes inconsistent or, (2) there is an inaccuracy in the data source. For example, instability in routing combined with a delay in obtaining routing data could cause inconsistency.

The impact of an address misclassification depends on the monitoring infrastructure and if the honeynets actively respond to incoming packets. For example, misclassifications that occur when using a blackhole route are likely due to operator error and would have happened regardless of a Dark Oracle deployment. However, if a system like *darktrap* is being used, a contention between live systems and honeypot systems can arise. If the address of a server is misclassified, then it is possible that a valid client could interact with a honeypot instead.

The simplest way to avoid misclassification is to minimize inconsistent state and inaccurate data sources. For example, by peering directly with border routers it is possible to minimize inconsistent state between the Dark Oracle and BGP data sources. Inaccurate data sources are often a result of misinformation so education and enforcing strict network policy can minimize inaccuracy.

Despite the best prevention efforts it is still possible to get misclassification. Two steps to reduce the impact are whitelists and less aggressive monitoring. It is possible avoid interactions between legitimate clients and honeypots by whitelisting critical servers. Another technique is to use less aggressive honeynets. For example, a passive capture system that is not inline with the network can be used instead of interactive honeypots for important subnets. Such a system prevents disruption to connectivity but still allows the collection of detailed data.



(a) (b) (c)

Figure 6: 2D visualization of IP address blocks in the (a) bogon list, (b) allocated by RIPE and ARIN, and (c) advertised via BGP as observed by all RouteViews peers on September 20, 2005. White space represents valid addresses and black space dark addresses and area is proportional to the amount of address space.

6 Dark Oracle Evaluation

In this section we evaluate the proposed architecture and the Dark Oracle prototype. The evaluation is divided into the three parts. In the first part, we use data from a regional ISP, a large enterprise, and an academic network to analyze the quantity, density, and stability of addresses produced by the external routing, internal routing, and host configuration data sources. In the second part, we deploy the *darktrap* and a *blackhole* route on a live network and evaluate the visibility provided by the Dark Oracle by comparing it with existing darknets. Finally, we analyze the effectiveness of using the addresses discovered by the Dark Oracle for detecting targeted and internal attacks.

6.1 Data Source Evaluation

In this subsection we analyze the addresses provided by the different data sources used in the Dark Oracle.

6.1.1 External Routing: BGP

We begin by comparing the BGP data source to similar sources of global Internet reachability information and investigate the stability of the addresses discovered over time. We use address allocations from the major regional registries and non-routable addresses from the bogon list [12] as two other major sources of Internet reachability data. To compare data sources we plotted a snapshot of the prefixes from each data set from September 20, 2005 using a 2D quadrant-based visualization technique that maps all IPv4 space onto a two-dimensional plane [27]. Unused address space is shown in black and used address space in white. Area in the plot is directly proportional to the amount of address space visualized, so a single /8 network takes up 1/256 of the area in each plot. A plot of the bogon list is shown in Figure 6(a); the allocation databases of the two largest regional registries, ARIN and RIPE, are shown in Figure 6(b); and all announced BGP prefixes from RouteViews [34] in Figure 6(c).

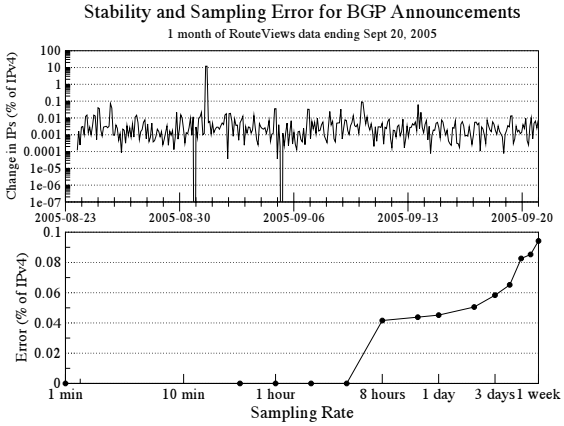


Figure 7: Change in addresses advertised through BGP over time as a percentage of 32-bit (IPv4) space. BGP advertisements observed by all RouteViews peers over one month ending September 20th, 2005.

Figure 6 shows how allocation information becomes successively more fine-grained as one moves down the allocation hierarchy. The figure also shows qualitatively how the more detailed information provided by the registries and then BGP reveal highly distributed dark addresses. Quantitatively, BGP also reveals the most dark addresses. The bogon list indicates 1,898,557,675 dark addresses, the combined regional registry data reveals 2,396,409,621 dark addresses, and the BGP data reveals 2,872,949,395 dark addresses.

Another question is the stability of the BGP data. That is, how often are addresses added or removed. Churn in BGP announcements is well-documented and although there are often a large number of update messages, we found that the relative amount of addresses that change

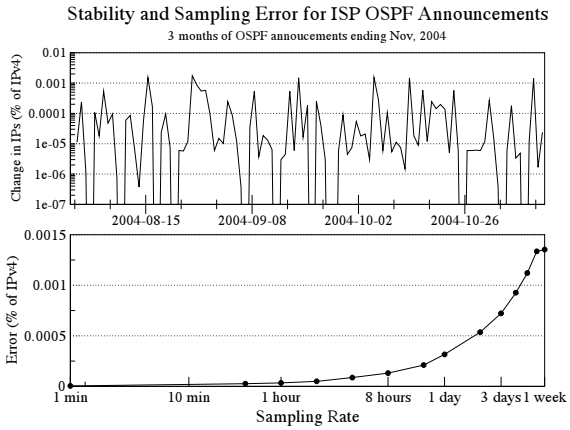


Figure 8: Absolute change in number of addresses advertised through OSPF over 3 months in late 2004 as a percentage of 32-bit (IPv4) space. Observed on the OSPF backbone of a regional ISP.

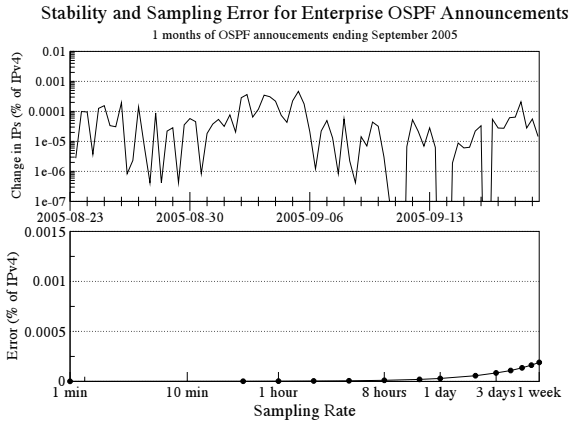


Figure 9: Absolute change in number of addresses advertised through OSPF over 1 month in September 2005 as a percentage of 32-bit (IPv4) space. Observed on the OSPF backbone of a large enterprise network.

is quite small. Figure 7 plots the absolute number of addresses that change as a percentage of all possible IPv4 space. We found address churn for BGP is typically between 0.01 and 0.001 percent of all IPv4 space (that's approximately a /16 in size) per 4-hour period.

We also evaluated the error incurred when sampling the BGP data sources. The sampling error is shown in Figure 7. Because the data from RouteViews is updated on a 4-hour basis, the error is zero up to 4 hours. The error with an 8 hour sampling period is 0.04%, which suggests the BGP data source should be updated more frequently. For example, having the BGP module in the Dark Oracle peer directly with the border routers would provide more accurate external reachability information.

6.1.2 Internal Routing: OSPF

To evaluate the use of IGP data for the Dark Oracle, we analyzed OSPF data captured at a large enterprise and a regional service provider. The large enterprise was allocated approximately 900,000 addresses by a regional registry, accounting for 0.02% of all IPv4 space. By analyzing the link state advertisements, we were able discover the number of addresses that were internally routable in a certain part of the network. Over a three-week observation period we discovered 112,423 addresses advertised through OSPF. Of these, 56,139 addresses were from private address space and 56,284 addresses were allocated by a regional registry.

The use of private address space in the enterprise is also very interesting. The 56,139 private addresses were from all three private prefixes (i.e., 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12) but only covered 0.3% of the total possible private addresses. This means a huge number of unused private addresses were available.

The mix of addresses observed through OSPF in the regional service provider was somewhat different from the enterprise. We observed 20,055,568 allocated and globally routable addresses, which is 0.47% of IPv4 space. Although this is much larger than the enterprise, only 512 addresses from private address space were advertised. This difference may stem from the operational goals of a provider and an enterprise. An enterprise primarily needs IP addresses for local reachability, especially when you consider the widespread use of proxies. On the other hand, a service provider, like the one we profiled, provides global Internet connectivity and thus globally reachable addresses are most important. These differences suggest that a service provider should consider constructing honeynets primarily from globally reachable addresses and an enterprise from large numbers of private addresses.

The addresses advertised through OSPF at the large enterprise and the service provider also showed good stability. Figure 8 shows the address churn at the regional service provider and Figure 9 shows the churn at the large enterprise. The average churn is approximately 0.00001% of IPv4 space per 8 hours. We also measured the error incurred by sampling the data source at different intervals. It turned out much of of the churn was due to the advertisement and withdrawal of a single /32 prefix so the sampling error remained small. Sampling at one-hour intervals produced very little error, so if the Dark Oracle was using OSPF data to interpret the passive output of a blackhole route it could poll the routers instead of participating in OSPF.

6.1.3 Host Configuration: DHCP

To evaluate the utility of the host configuration data source in the Dark Oracle we analyzed the number and stability of dark addresses provided by a DHCP server. We used data from a DHCP server deployed in a department in a large academic network. The DHCP server

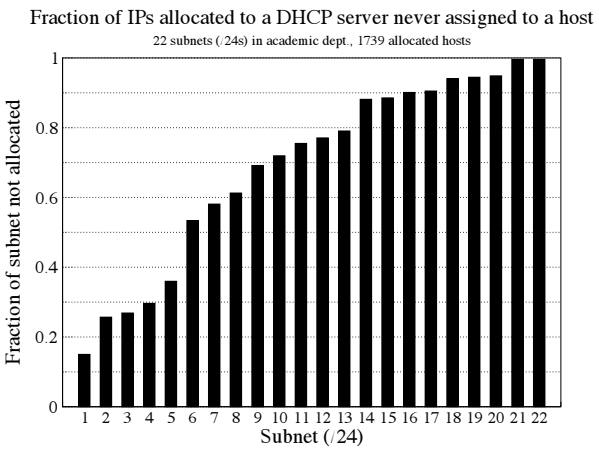


Figure 10: Amount of address space allocated to a DHCP server by /24 subnet in a department of a large academic organization that was never assigned to a host during September, 2005. In total, 70% of the addresses allocated to the DHCP server were never used.

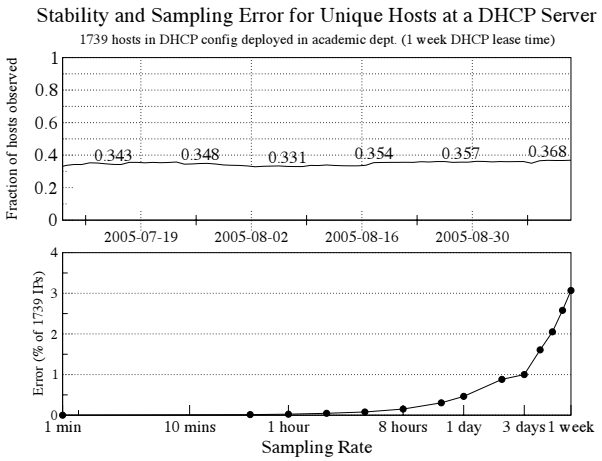


Figure 11: Fraction of unique hosts in the DHCP server configuration file that obtained or renewed an address over time. The average number of hosts active at any one time is approximately 35%. Data over two months in a department at a large academic organization during September, 2005. The DHCP server was configured with 1 week lease times.

configuration file included 1,802 static entries to allocate addresses based on MAC address.

The DHCP server was assigned a total of 22 /24 subnets from which the 1802 hosts were allocated an IP address. This means that 3319 addresses were never used. The distribution of these unused addresses by subnet is shown in Figure 10. 16 of the 22 subnets were more than 50% unused leaving a large number of dark addresses. Equally interesting, the subnet with the most hosts was still left with 15% of the space unallocated.

We also tracked the amount of time each host was active by monitoring when hosts were assigned or renewed a DHCP leases from the server. Figure 11 shows the number of addresses used over two months. Surprisingly, only about 35% of the 1,802 addresses were in use at any time and the usage was very stable (the DHCP server was configured with a 1-week lease time which likely improved stability). To put this in context, if we were to just use OSPF data, we would observe the 22 subnets allocated to DHCP and assume all 22 were used. But, by using host configuration data we were able to discover that only about 631 addresses out of the possible 5,566 usable addresses were in use.

We also looked at the sampling error incurred by updating the DHCP data source less frequently. As shown in Figure 11, the mean sampling error remains well under 1% for almost 3 days. This is partially related to the long lease time (one week), but also indicates the Dark Oracle could sample much less frequently and maintain almost perfectly in-sync.

Finally, one might expect some hosts connecting through DHCP to come and go with high frequency. We also analyzed how long an address that was newly classified as dark stayed dark. Over the entire evaluation period we found the mean time an address was classified as dark was 8.85 days and the median was 18.02 days. Thus, a newly dark address will typically stay dark for at least two weeks, although certain addresses fluctuate more rapidly (perhaps due to mobile users).

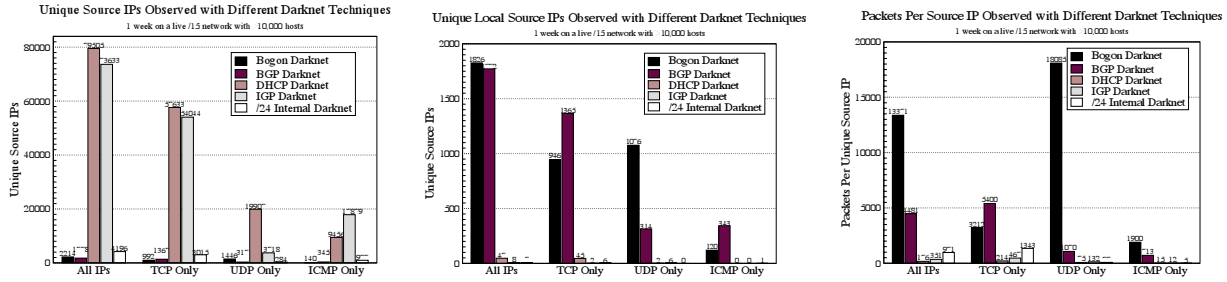
6.2 Live Deployment Results

In this subsection we evaluate a live deployment of the Dark Oracle on a real network in a large academic institution. The system was deployed at a central campus router serving approximately 10,000 unique hosts in two /16 networks.

To redirect traffic to our honeynet, we used the *darktrap* program and routing blackhole described earlier. *darktrap* was used to capture traffic to dark addresses discovered by the BGP and host configuration modules, and a routing blackhole was used to capture dark addresses in OSPF. *darktrap* was executed on a 3Ghz system and input traffic was from an optical tap from a span port off a Cisco Catalyst 6500. Traffic destined to the routing blackhole was forwarded to an interface on the same box and integrated with the dark traffic.

6.2.1 Addresses Discovered

Before looking at what was detected, we review the number of dark addresses discovered by the Dark Oracle deployment. The number of prefixes, dark addresses, and total fraction of IP space that was dark for each data source is shown in Figure 12. The fraction of address space that was dark for each data source was computed



(a) (b) (c)

Figure 13: Results from a one week deployment of the Dark Oracle on a large academic network serving approximately 10,000 hosts. Each graph shows the result from the BGP, IGP, host configuration Dark Oracle components. A single /24 darknet is provided for comparison with traditional honeynet monitoring approaches. (a) shows the number of unique source IPs detected, (b) shows the number of unique IPs from within the academic institution address space detected, and (c) shows the number packets per unique source IP.

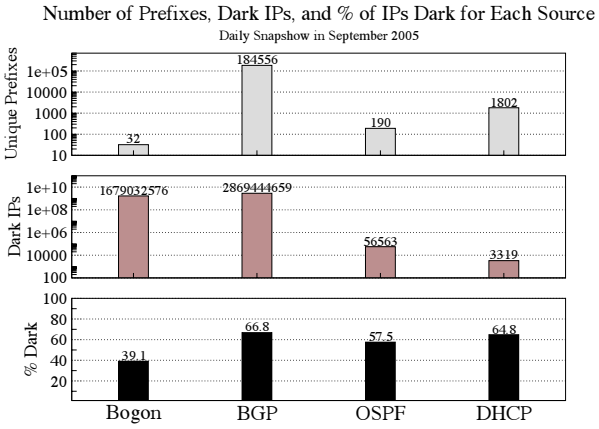


Figure 12: The number of prefixes, dark addresses, and total fraction of IP space dark captured with a snapshot of each Dark Oracle data source on a day in September 2005 in a large academic network.

by taking the number of unused addresses over the total number of addresses managed by the data source. For example, there were 5,120 total addresses allocated to the DHCP server and out of those, 1,801 addresses were configured to be used by the server. An interesting result shown in Figure 12 is that more than 50% of the addresses in the external and internal routing and host configuration sources were dark.

6.2.2 Honeynet Detection Results

To evaluate the utility of the addresses discovered by the Dark Oracle we now characterize the traffic captured by *darktrap* and the blackhole route. The metric we use is the number of unique source addresses observed. The number of unique source IPs provides a first-order approximation of the number of unique infected/misconfigured hosts. We make no attempt to separate misconfigured hosts from infected hosts as both provide important information from the perspective of network operators. Furthermore, existing signature-based and prevalence-based detection systems can be used to help identify malicious traffic [25].

We now present results from a one week deployment of the Dark Oracle on a large academic network. The results are shown in Figure 13. For comparison, we also include results from a single statically allocated /24 darknet and a darknet composed of only bogon addresses operating during the same time period within the same academic network.

Source IPs: Figure 13(a) shows the number of unique source IPs detected at the dark addresses discovered using different Dark Oracle data sources. The data is separated by IP protocol. UDP source addresses are sometimes spoofed but the source address on TCP packets are most often valid in order complete the handshake.

The huge number of IPs detected by the IGP and host configuration data sources indicates the importance of having both breadth and good placement. The DHCP data source observed almost 13 times more addresses than the single /24 darknets. Recall that the IGP and host configuration data sources can capture attacks coming into the network. Thus, the almost 80,000 source IPs detected are likely externally-sourced attacks coming into the network. In contrast, the few thousand IPs detected by the bogon and BGP data sources are likely hosts on the same network.

Local Source IPs: To evaluate the locality of the detection results we plotted only those source IPs that were within the address space of the academic network. The results shown in Figure 13(b) indicate that the addresses from the bogon and BGP data sources detected locally infected/misconfigured hosts while the IGP and DHCP data sources revealed external hosts.

Destinations Per Source IP: The bogon and BGP data sources provide addresses for *outgoing* honeynets and thus information on infected/misconfigured hosts from inside the network. However, the bogon and BGP

data sources also reveal many more addresses than the other data sources so we would expect those addresses to capture a higher percentage of the packets from each infected/misconfigured host. Figure 13(c) plots the average number of packets sent by hosts detected with addresses from each data source. As expected, the bogon and BGP data sources provided addresses that have a higher probability of detecting a local host and thus are well-suited for local detection.

6.2.3 Classification Error

To track the number of misclassifications made by the Dark Oracle we wrote a program called *addrmon* that monitored the same router span port as *darktrap* and flagged an IP address as active if it observed that address sending an IP packet. Throughout the entire week-long period we observed 11,118 active IPs on the network. 45 of those IPs were classified as dark by the Dark Oracle (we removed those addresses from our analysis). It is also important to note that we just looked for a single packet so some of those 45 addresses could have been spoofed, and thus were actually dark. Further investigation of those addresses revealed that they were nearly all statically configured hosts.

6.3 Detecting Targeted Attacks

We have shown how the Dark Oracle provides many dark addresses but equally or more important, those addresses are highly distributed throughout the network. We now evaluate how the distributed property of these addresses provides visibility into targeted attacks that would be missed by existing contiguously allocated honeynet systems. Because the addresses are located in many different subnets, honeynet sensors can be pervasively deployed in hundreds or thousands of different parts of the network near to production systems and critical network assets.

To evaluate the importance of having distributed dark addresses we now analyze the time small but well-placed sensors take to detect different targeted attacks. We model an intelligent attacker that has knowledge of which subnets contain vulnerable hosts. Our model is based on botnet scanning behavior which we empirically demonstrated in Section 3. Thus, rather than scanning the entire IPv4 address space the attacker will chose a specific subset like a /24 or /16 to scan.

A random scan of IP space is a straightforward process to model. Previous work has looked at the question of how big a darknet needs to be to detect a random scanning worm with a certain confidence [21]. We can take that understanding and extend it to understand targeted scan detection. Moore *et al.* [21] found that the probability of observing one or more packets from a host with a random scan rate r using a detector with coverage p after time T is given by $P(t \leq T) = 1-(1-p)^{rT}$. They also found that the amount of time T needed to assure a certain probability Z of detecting at least one packet from a scanning host is given by $T = \frac{-1}{r \log_{\frac{1}{Z}}(1-p)}$.

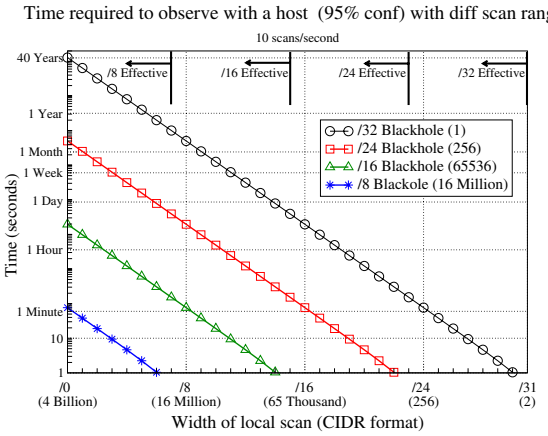


Figure 14: Time required to observe with 95% confidence a packet from a host randomly scanning different ranges of addresses with 4 different sized darknets.

In Figure 14, we plot the detection rate with darknets of different sizes as a function of the width of a targeted scan using the above equation. This model the time needed to assure a 95% confidence of detecting a packet from a scan of a certain number of addresses using a darknet having a certain number of addresses. For example, it takes about one minute to detect a packet from a /16 (65,536 addresses) scan with 95% confidence using a /24 (256 addresses) darknet sensor located within the scan range. Detecting a packet from the same scan with the same confidence using a /32 (a single host) would take 5.5 hours. Also, the same local /16 scan could not be detected by a /16 or /8 sensor, which are too large so they are simply not applicable.

The surprising result of this analysis is that even a darknet covering a single address in the right place is an effective tool at detecting targeted scanning behavior. Highly-distributed dark addresses from the Dark Oracle provided by data sources like DHCP and BGP therefore provide the capability to quickly detect targeted incoming and outgoing scans from botnets and other threats.

7 Limitations and Future Work

We wrap up our discussion of the Dark Oracle by discussing possible limitations of the system, describing other novel data sources that could be used to enhance visibility, and detailing how data from different organizations could be combined to construct a powerful, globally-scoped system.

One limitation in deploying a system like the Dark Oracle is the need for access to host configuration data sources. Real networks are complicated and there are often machines that are not in common allocation databases. For example, data centers often have systems with statically configured addresses and many departments manage addresses differently. Informing the Dark Oracle about statically configured machines or getting access to host configuration information in certain parts of the network may not be practical.

Another limitation is address misclassification due to data source instability or inaccuracy. We discussed this issue in Section 5.6 and related several preventive measures to mitigate risk.

There is also the possibility that an attacker could *fingerprint* the dark addresses and attempt to avoid them. Beyond the simple defense of making the honeynets act as much like real systems as possible, the huge range of dark addresses discovered by the Dark Oracle provides strong defense. For example, it is possible to respond with honeypots from IPs that randomly rotate based on the source IP of the attacker. Such simple defenses render algorithms like probe response attacks far more difficult to execute [5]. Even with a complete map of dark addresses, it is impractical to encode them in self-propagating malware like worms due to payload size constraints [38].

The flexibility that makes the Dark Oracle resistant to fingerprinting also makes it very expandable. Because the data sources used in the Dark Oracle are independent, it is simple to deploy the Dark Oracle in stages and add new data sources as needed. There are many data sources that provide allocation data with other interesting perspectives. For example, dark addresses in the address blocks assigned to VPN servers, addresses blocked by network-based and host-based firewalls, and even ACL violations in routers.

One promising pool of dark addresses that could be used with the Dark Oracle is unused TCP and UDP ports. The live computers sitting around a network are often idle and have many unused TCP and UDP ports. A daemon running on each end host could inform the Dark Oracle about these unused ports and packets destined to these unused ports could instead be forwarded to a honeynet.

As a preliminary investigation of the idea of monitoring unused ports we measured the mean number of ports that were used per 5 minutes per local source IP address in the large enterprise and academic network. As Figure 15 shows, there are many unused ports that could be leveraged. Hosts on the academic network used less then 1,000 ports on average which is far less then the possible 65,335 ports. The spikes in the enterprise data are interesting and are likely correlated with backup activity.

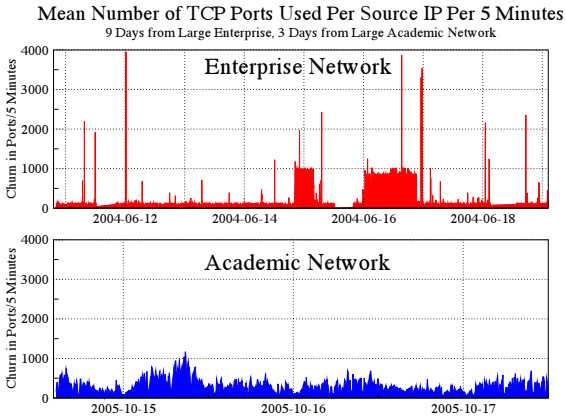


Figure 15: Mean number of TCP port unused per unique local source IP per 5 minutes on a large enterprise and large academic network over a few days. Measurements done in June 2004 and October 2005.

Another interesting research area lies in sharing the allocation data between organizations to improve global visibility. Previous work has looked at sharing dark addresses between an ISP and its customer [16], but it is also possible to connect Dark Oracle instances together to form a global network of fine-grain dark address information services. This would enable organizations to construct much more robust outgoing filtering devices.

8 Conclusion

This paper has introduced the Dark Oracle, a system that automates the process of discovering unused and unreachable addresses inside a network. We described a general architecture that integrates external routing data like BGP, internal routing data like OSPF, and host configuration data like DHCP server logs to construct a locally-accurate map of dark addresses. We experimentally evaluated the Dark Oracle using data from a large enterprise network, a regional ISP, and deployment of the Dark Oracle on a large academic network. We showed how the Dark Oracle provided addresses that revealed almost 80,000 unique source IPs compared to 4,000 with a traditional /24 darknet. We also demonstrated how the unique perspective of Dark Oracle provided visibility into internal threats and targeted attacks. Finally, we described future work and extensions to the Dark Oracle such as leveraging unused TCP and UDP ports on live hosts and combining many Dark Oracles to construct a global dark network.

Acknowledgments

This work was supported by the Department of Homeland Security (DHS) under contract number NBCHC040146, and by corporate gifts from Intel Corporation and Cisco Corporation.

References

[1] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis. Detecting targeted attacks using shadow honeypots. In *Proceedings of the 14th USENIX Security Symposium*, Baltimore, MD, August 2005.

[2] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. The Internet Motion Sensor: A distributed blackhole monitoring system. In *Proceedings of Network and Distributed System Security Symposium (NDSS '05)*, San Diego, CA, February 2005.

[3] Michael Bailey, Evan Cooke, Farnam Jahanian, Niels Provos, Karl Rosaen, and David Watson. Data Reduction for the Scalable Automated Analysis of Distributed Darknet Traffic. *Proceedings of the USENIX/ACM Internet Measurement Conference*, October 2005.

[4] Michael Bailey, Evan Cooke, David Watson, Farnam Jahanian, and Jose Nazario. The Blaster Worm: Then and Now. *IEEE Security & Privacy*, 3(4):26–31, 2005.

[5] John Bethencourt, Jason Franklin, and Mary Vernon. Mapping Internet sensors with probe response attacks. In *Proceedings of the 14th USENIX Security Symposium*, Baltimore, MD, August 2005.

[6] Bill Cheswick. An evening with Berferd in which a cracker is lured, endured, and studied. In *Proceedings of the Winter 1992 USENIX Conference: January 20 — January 24, 1992, San Francisco, California*, pages 163–174, Berkeley, CA, USA, Winter 1992.

[7] Computer Associates. Win32.Agobot. `http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=37776`, July 2004.

[8] Evan Cooke, Michael Bailey, Z. Morley Mao, David Watson, and Farnam Jahanian. Toward understanding distributed blackhole placement. In *Proceedings of the 2004 ACM Workshop on Rapid Malcode (WORM-04)*, New York, Oct 2004. ACM Press.

[9] Evan Cooke, Farnam Jahanian, and Danny McPherson. The Zombie roundup: Understanding, detecting, and disrupting botnets. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2005 Workshop)*, Cambridge, MA, July 2005.

[10] Evan Cooke, Z. Morely Mao, and Farnam Jahanian. Hotspots: The root causes of non-uniformity in self-propagating malware. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'2006)*, June 2006.

[11] Team Cymru. The darknet project. `http://www.cymru.com/Darknet/index.html`, June 2004.

[12] Team Cymru. The Bogon List. `http://www.cymru.com/Documents/bogon-list.html`, June 2005.

[13] Warren Harrop and Grenville Armitage. Greynets: A definition and evaluation of sparsely populated darknets. In *Proceedings of the ACM SIGCOMM MineNet Workshop*, Philadelphia, PA, August 2005.

[14] Internet Assigned Numbers Authority (IANA). Internet Protocol V4 Address Space. `http://www.iana.org/assignments/ipv4-address-space`, June 2005.

[15] Xuxian Jiang and Dongyan Xu. Collapsar: A VM-based architecture for network attack detention center. In *Proceedings of the 13th USENIX Security Symposium*, San Diego, CA, USA, August 2004.

[16] Balachander Krishnamurthy. Mohonk: Mobile honeypots to trace unwanted traffic early. In *Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*, pages 277–282. ACM Press, 2004.

[17] Abhishek Kumar, Vern Paxson, and Nicholas Weaver. Exploiting underlying structure for detailed reconstruction of an internet-scale event. *Proceedings of the USENIX/ACM Internet Measurement Conference*, October 2005.

[18] Craig Labovitz, Abha Ahuja, and Michael Bailey. Shining Light on Dark Address Space. `http://www.arbornetworks.com/`, November 2001.

[19] McAfee. W32/Sdbot.worm. `http://vil.nai.com/vil/content/v_100454.htm`, April 2003.

[20] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. Inside the Slammer worm. *IEEE Security & Privacy*, 1(4):33–39, 2003.

[21] David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage. Network telescopes. Technical Report CS2004-0795, UC San Diego, July 2004.

[22] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet denial-of-service activity. In *Proceedings of the Tenth USENIX Security Symposium*, pages 9–22, Washington, D.C., August 2001.

[23] Richard Mortier. Python routeing toolkit. *IEEE Network*, 16(5):3–3, September 2002.

[24] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of Internet background radiation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40. ACM Press, 2004.

[25] Vern Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[26] Niels Provos. A Virtual Honeypot Framework. In *Proceedings of the 13th USENIX Security Symposium*, pages 1–14, San Diego, CA, USA, August 2004.

[27] S. Qiu, Patrick McDaniel, Fabian Monrose, and Avi Rubin. Characterizing address use structure and stability of origin advertizement in interdomain routing. Technical Report NAS-TR-0018-2005, Pennsylvania State University, July 2005.

[28] Colleen Shannon, David Moore, and Jeffery Brown. Code-Red: a case study on the spread and victims of an Internet worm. In *Proceedings of the Internet Measurement Workshop (IMW)*, December 2002.

[29] Dug Song, Rob Malan, and Robert Stone. A snapshot of global Internet worm activity. FIRST Conference on Computer Security Incident Handling and Response, June 2002.

[30] Lance Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2002.

[31] Lance Spitzner et al. The honeynet project. `http://project.honeynet.org/`, June 2004.

[32] Symantec Corporation. DeepSight Analyzer. `http://analyzer.securityfocus.com/`, 2005.

[33] Johannes Ullrich. DShield. `http://www.dshield.org`, 2000.

[34] University of Oregon. RouteViews project. `http://www.routeviews.org/`.

[35] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Scalability, fidelity and containment in the Potemkin virtual honeyfarm. In *Proceedings of the 20th ACM Symposium on Operating System Principles (SOSP)*, Brighton, UK, October 2005.

[36] Jianhong Xia, Lixin Gao, and Teng Fei. Flooding Attacks by Exploiting Persistent Forwarding Loops. *Proceedings of the USENIX/ACM Internet Measurement Conference*, October 2005.

[37] Vinod Yegneswaran, Paul Barford, and Dave Plonka. On the design and use of Internet sinks for network abuse monitoring. In *Recent Advances in Intrusion Detection—Proceedings of the 7th International Symposium (RAID 2004)*, Sophia Antipolis, French Riviera, France, October 2004.

[38] Cliff C. Zou, Don Towsley, Weibo Gong, and Songlin Cai. Routing worm: A fast, selective attack worm based on IP address information. Umass ECE Technical Report TR-03-CSE-06, November 2003.