Early Detection of Internet Worm Activity by Metering ICMP Destination Unreachable Messages

George Bakos and Drs. Vincent H. Berk

ISTS Dartmouth College, Hanover, NH, USA

ABSTRACT

Early warning of active worm propagation over the Internet is of vital importance to first responders. Knowing an active worms characteristics very early in its propagation can significantly reduce the damage it may cause.

In this paper we propose an early warning system that uses ICMP Destination Unreachable (ICMP-T3) messages to identify the random scanning behavior of worms. Participating routers across the Internet send Blind Carbon Copies of all their locally generated ICMP-T3 messages to a central collection point. Incoming messages are abstracted and patterns identified. Using the methods discussed in this paper we identify "blooms" of activity that are a clear signature of worm propagation.

Preliminary test results have shown that actively spreading worms can be identified in the first few minutes after they are launched. By using the characteristics gathered in those early stages, action can be taken and widespread damage might be avoided.

Keywords: ICMP, Destination Unreachable, Worm, Early Warning, ISTS, Dartmouth College

1. INTRODUCTION

This research project seeks to determine the feasibility of analyzing ICMP unreachable messages generated by routers throughout the Internet, to detect large-scale scans indicative of worm propagation, as well as other significant changes in pseudo-random target scan patterns. By correlating detected scan events from a statistically significant number of participant routers worlwide, it is hypothesized that scan patterns will emerge indicative of high-priority events.

Data gathered during and after Nimda[1] worm analysis shows that the vast majority of responses to connection attempts were ICMP Type 3 (ICMP T3), unreachable, messages. Tables 1, 2 and 3 demonstrate these percentages. Note that Nimda (table 1) used a random target selection algorithm weighted towards IP addresses numericly close to the source address. The ping and tcp port 80 scans (tables 2 and 3) used a non-weighted randomization scheme that avoided IANA reserved address space as determined via the unix command: whois iana@whois.arin.net.

The embedded content of the ICMP messages contains sufficient details of the failed datagram to ascertain the intended purpose of that datagram. A large number of these with similar attributes could be assigned to a unique scan event, or "bloom", and correlated against similar observations. This correlation process, taking place in near-realtime, will reveal new large-scale scan and worm activity.

Further author information:
G.B.: E-mail: gbakos@ists.dartmouth.edu
V.H.B.: E-mail: vberk@ists.dartmouth.edu
Telephone: 1 603 646 0700, Address: 45 Lyme Rd. Suite 200, Hanover, NH 03755, USA

Table 1: Target selection scan events - nimda host scan

syn	18369	
ack/syn	506	8.60 %
rst	519	8.82 %
icmp t3	4858	82.57 %

Table 2: icmp ping requests (type 8)

ping requests	324128	
ping replies	6098	17.10 %
icmp t3	29557	82.89 %

Table 3: tcp port 80 connections - non-nimda

syn	1981444	
ack/syn	9374	6.17 %
rst	16369	10.79 %
icmp t3	125946	83.02 %

2. TECHNICAL BACKGROUND

2.1. Overview of system compromise

Historically, unauthorized remote access to computer was the culmination of several fairly well defined phases, closely paralleling military operations: (Terms and phases from 2.)

1. Reconnaissance - the attacker gathers addreses of systems that are responsive. Often, particular services are targeted to see if they are offerred on a target. 2. Probe and Attack - As the attacker is limited in the types, and particular versions, of services that he is capable of using to his advantage, he needs to identify the running processes on these accessible systems. This can be done by manual examination, automated retrieval of banner information, observation of system responses to various stimuli, etc. 3. Toehold - Initial access is gained. Most system compromises are the result of poor configuration of publicly accessible services, or programming flaws that permit unchecked data to be accepted as input, potentially allowing this unchecked data to overwrite existing data structures and execute instructions on the target system. These are commonly known as "buffer overflows."[3] 4. Advancement, 5. Stealth, 6. Listening post, 7. Takeover - Depending on the ultimate goals of the intruder, the system may be patched against further compomise, additional local-style attacks may be used to acheive a higher level of priviledge, backdoors installed to facilitate subsequent returns of the attacker or others, software downloaded and/or installed to be used in further attacks, or there could be destruction, further network compomise, modification or access to sensitive or valuable data.

More recently, the trend has shifted towards greater automation of some, if not all, of the above phases of attack. Many of these composite tools are referred to, in the attack community, as autorooters.[4] However, one component remains observable and fairly recognizeable: broad scanning. Unless the attacker is already intimate with his target environment, there must be some method of identifying systems that would be receptive to the attack sequence. Automated attack "scripts" must first attempt to connect via a tcp SYN, ACK-SYN, ACK , or "three-way handshake",[5] before data can be sent. In the case of UDP communication, there is no such negotiation; data is merely delivered to every address in the target range and the attacking process listens for the desired replies. Still another variant, slightly more stealthy, is using abnormally configured packets to elicit known responses to these stimuli. An initial FIN, or finish packet, for instance, will trigger a RST if the service is not available, but there will be no response at all if the daemon is listening.[6] When any of these

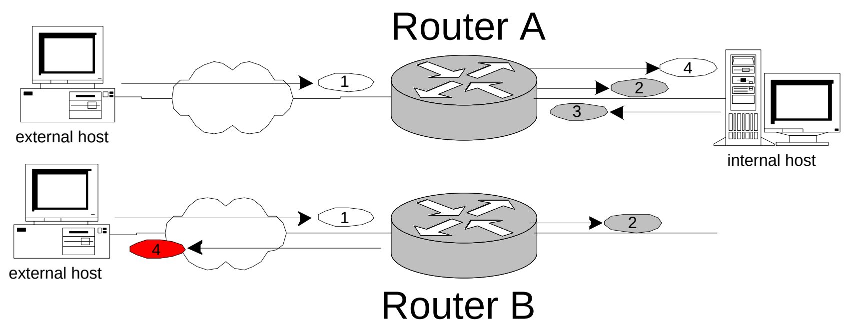

Figure 1: Example of ARP communication.

techniques are used against large numbers of IP addresses, there is considerable liklihood of individual scan detection using conventional perimiter defense system log analysis. This model, however, breaks down rapidly if the attacking system is randomizing target IP addresses, minimizing the chances that his activities will fall upon any particular network with sufficient frequency to trigger counter-threshold based scan detectors.

2.2. Recent Worm Activity

Self-propagating malicious programs, moving amongst interconnected computer systems, have come to be commonly referred to as worms. Although there have been numerous email-borne self replicating virii, also called "worms", in this paper we will use the term "worm", as in (7) to describe malicious programs that are completely autonomous, infecting and replicating without requiring any action on the part of the system operator.

The most dangerous worms to date have been those that exhibit explosive growth rates, far outpacing efforts to stem their activity. CodeRed (CR) and CodeRed 2 (CRv2) took advantage not only of of programming flaws in Microsoft Internet Information Services (IIS), but also of default installation settings. In order to facilitate rapid propagation, CR used a random number generator process as its target selection algorithm. Initially, the random number generator sufferred from a static seed, thus every new copy of the worm scanned the same IP space over and over. On July 19th, the architect (or another party interested in improving the code) modified CR to use host time as a pseudo-random seed. With this enhancement, more than 359,000 computers were infected in under 12 hours[8] The payload was particularly mild, however: a simple time-delayed Distributed Denial of Service (DDOS) attack directed against whitehouse.gov's IP address.

As it is highly unlikely that the CR worm architect was technically unable to build a more lethal DDOS weapon*, is is reasonable to assume that our success in dodging that bullet was partly due to the generosity of the worm programmer. More importantly, the rapid response of the security community, in particular Marc Maiffret and Ryan Permeh of eEye Digital Security's analysis,[9] the SANS Institute's information dissemination channels,[10] and many others, aided the effort to slow the infection rate and patch vulnerable systems. Please note, this community effort was initiated only after notification by administrators of infected or scanned systems, and was not heeded by a large number of systems administrators, resulting in the rapid propagation of CRv2.[11] Rapid response is essential in limiting the spread and global impact of active worms. ICMP unreachable messages may provide this notification.

2.3. Internet Control Message Protocol behavior

Aside from being an occasoinal covert communications protocol,[3] ICMP provides essential notification for network problems. Host or net resources not being reachable, TTL values decrementing to zero, route redirection,

*One of the measures taken to defend www.whitehouse.gov, was to change its IP address. gethostbyname() is a commonly used C programming language function to resolve IP addresses from hostnames. Use of it in conjunction with the domain name, rather than its IP address, would have rendered any IP address manuevers less effective.

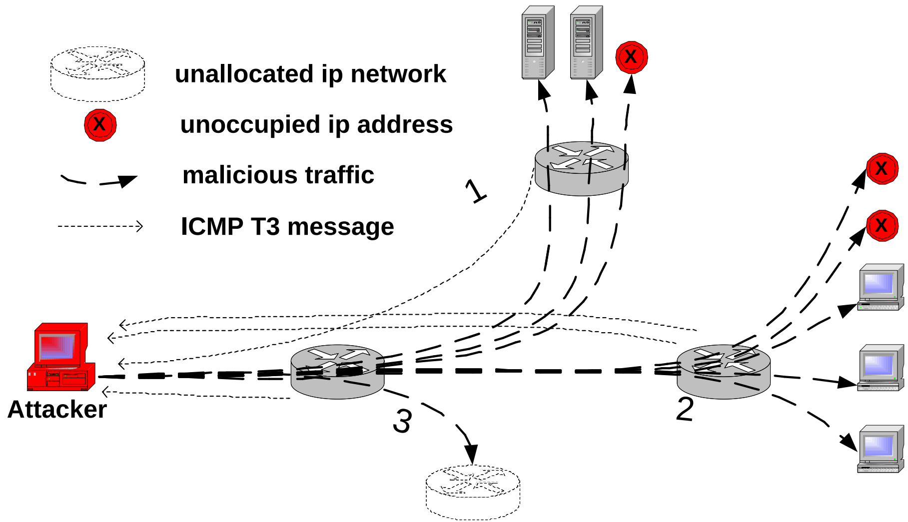

Figure 2: Active worm scan activity.

source quench; these are normal, essential ICMP message types.[5] In particular, we are concerned with messages produced by routers as a result of unsuccessful traffic.

When a datagram is presented to a router for delivery, the logical IP address is encapsulated within a physical delivery frame. This frame is specifically addressed to the hardware address of the router's interface. In Figure 1, upon successful receipt of the frame (1), router A's stack discards the frame encapsulation, makes the appropriate routing decision and prepares to transmit it on the appropriate interface to its destination. In order to construct a frame header with the correct datalink (MAC) header, the router first queries the network (2) by broadcasting an Address Resolution Protocol (ARP) address request. If the host is available, it will ARP reply (3) with its IP address. The router can then construct the frame header and deliver the packet (4).[12]

In the case of figure 1, Router B, the host is not available, thus it never receives a reply to its ARP request (2). The router then informs the sender that the destination is not available by encapsulating a portion of the original datagram in an ICMP Type 3 (ICMP-T3) message.[13] The amount of encapsulated data varies between various implementations, with a minimum requirement set in RFC792 of 28 bytes: the original IP header plus 8 octets of higher layer data. In the case of a TCP segment, these 8 octets would contain the source and destination ports, as well as the sequence number of the segment.[5]

These protocols should continue to operate as directed in the face of hostile traffic, as well. As mentioned earlier, unless the attcker is already familiar with the hosts he wishes to attack, as in "hitlist" scanning,[14] an intruder-to-be will attempt some measure of communications with a large number of target IP addresses, using etiher a sequential or pseudo-random selection process. Figure 2 describes such an event. The networks are populated to varying degrees, each of which presenting different target opportunities to the attacker. In cases 1 and 2, there will be, from the stub routers immediately responsible for those networks, ICMP host unreachable messages returned to the attacker, as we have seen above. Case 3, however is considerably different, and accounted for a number of failures in Cisco 12000 series router operating systems during CR and Nimda worm outbreaks.[15] As the entire network that the packets are destined for is unreachable, the dynamically configured router upstream of the attacker is aware via Border Gateway Protocol (BGP)[16] that these are non-routable packets. It will immediately create the ICMP-T3 message with a different code indicating that the failure is due to an unreachable network, rather than unreachable host.

3. HYPOTHESIS AND DESIGN

3.1. Hypothesis

By collecting voluntarily provided duplicate, or Blind Carbon Copy (BCC), ICMP T3 messages and correlating key elements of the encapsulated data, patterns can be observed indicative of scan activity. Through abstraction of these patterns and near real-time correlation of these abstractions, large-scale progressive scan events, such as worm propagation, become evident.

Table 4: Data from ICMP T3 embedded payload

Timestamp at Collector	Source address:port	Dest address:port	Protocol	Additional flags and values
12:23:43.926469	10.22.62.23:4237	10.10.10.172:80	tcp	DF, ttl 62, id 6610, len 60
12:23:44.266399	10.22.62.23:4386	10.10.10.173:80	tcp	DF, ttl 50, id 55000, len 60
12:23:47.426366	10.22.62.23:4398	10.12.12.246:80	tcp	DF, ttl 62, id 62755, len 60
12:24:24.736261	10.22.62.23:4425	10.13.13.234:80	tcp	DF, ttl 62, id 16746, len 60
12:24:49.196320	10.22.62.23:4429	192.168.1.223:80	tcp	DF, ttl 49, id 54395, len 60
12:25:01.422156	10.182.0.139:1304	172.19.45.75:80	tcp	DF, ttl 62, id 11806, len 60

By simple comparison that the first 5 events, although received at different times, are closely related. The single source address 10.22.62.23, attempted to send tcp packets to port 80 of the hosts listed. Other elements, such as TTL, length, and IP flags are similar. We can identify this event as a "bloom". The sixth message differs from the first bloom only in its source address. This is a significant alignment and may be indicative of common tool or technique usage, either automated or manual. This may be a client application (Netscape, email client, etc), scanning utility such as nmap[6] or a custom application like an automated worm.

There will be a high incidence of unique, non-correlated events, or "background noise". By carefully ordering events by priority, background noise can be effectivelly reduced so that system resources are conserved for the active tracking process.

New worms will show different patterns, which should become recognizable by their scanning behavior. Correlation engines will be capable of identifying these new patterns early in their life cycle. Multiple blooms with similar signatures, across any time frame, will be regarded as related ("compound blooms") and subsquently will be tracked more aggressively.

Discounting network latencies, redundant hostile connection attempts, perimiter defenses, and other quenching factors, an active worm, using pseudo-random target selection, should exhibit nearly exponential bloom rate expansion. This was observed during Code Red propagation rate measurements[8, 11] and is represented in figure 3. Future hostile events, using pseudo-random target selection mechanisms, will display similarly observable characteristics. Using ICMP T3 BCC analysis, we should detect such behavior at, or near, its onset.

3.2. Design

When a router is not able to reach a requested destination it can choose to reply to the sender with an ICMP Destination Unreachable (ICMP-T3) message.[13] As mentioned earlier, during worm propagation many addresses are scanned, either before or during the actual propagation. Since these addresses are generated in some random way many addresses are probed that are not actually associated with a machine.[14]

Since addresses are scanned all around the globe, many many routers will be involved to reach all the probed destinations. In order to detect this behavior a copy of all these ICMP-T3 messages needs to be brought together at a collection point. (Since this router is part of the communication it can generate a second copy of the message without having to "sniff" the network.) This 'Blind Carbon Copy' (BCC) can be generated by the router whenever a destination is unreachable and sent to a well known collection point. If the site policy is to

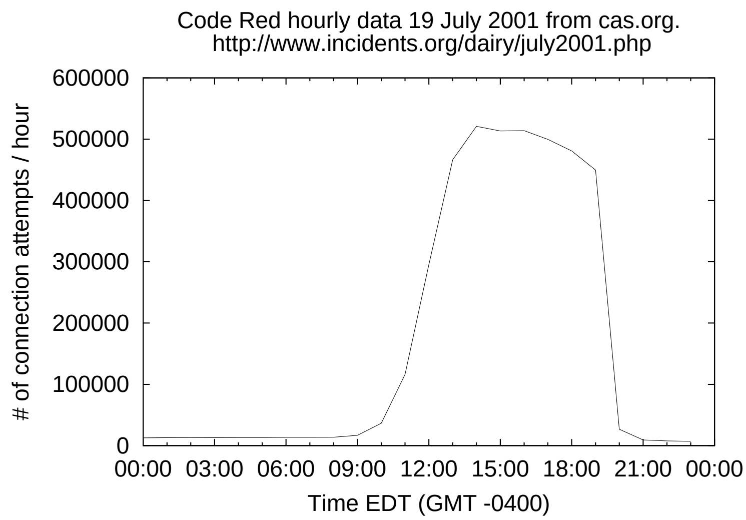

Figure 3: Code Red hourly connection attempts

block the return ICMP-T3 the router can still send a BCC to the collection point to notify of the connection attempt.

As described by Mansfield (17), communication networks are continously increasing in size. Since most worms will scan hosts at random, instead of scanning all the hosts on a particular remote network usually no more than one connection attempt is seen at any one time. This way a single network probe (possibly responded to by an ICMP-T3) can be easily overlooked in the vast amounts of data moving past. Network scans are becoming more and more frequent[4] and thus by themselves are not sufficient reason to investigate. This could mean that an infected host might go unnoticed.

Subsequenly, more advanced methods are being used to scan the Internet for possible targets. Methods include Hitlist Scanning, Topological Scanning and Permutation Scanning as described by Weaver (14). All of these technologies reduce the number of connection attempts that are performed to improve the worms performance. More efficient delegation of target address space results in less scanning and subsequently faster selective scans.

By having numerous routers on the Internet send their ICMP-T3 to a dedicated collector it can be easily detected that a single host is scanning across many networks. The ICMP-T3 messages arrive from many different locations and using the embedded packet in the message, knowledge can be derived regarding the source of the probe and possibly its intent.

By comparing the embedded original packet from the various ICMP-T3 messages coming in, similarities can be found. The embedded packet contained in the data portion of the ICMP-T3 message is at least the Internet Header and 8 bytes of the original datagram.[13] This information can be used to correlate several ICMP-T3 messages together as well as distinguish between several events happening at the same time. Relevant fields include:

- Source IP Address
- Destination IP Address
- Transport Protocol
- UDP/TCP Source Port

- UDP/TCP Destination Port
- ICMP Message Type and Code

Obiously, large numbers of packets coming from the same source IP address going to many different destinations, attempting a connection on any single port, should be considered suspicious.

3.3. Patterns

By analyzing and ordering the incoming ICMP-T3 messages it is possible to detect several significant events efficiently and fast. It is however important that a number of routers on the Internet will send their ICMP-T3 messages to the central collector. It is obvious that as the number of reporting routers grows the detection will be faster and more accurate. The following events will are observable (with a differentiation made on priority):

- Worm Propagation (High Priority)
- Host or Network loses Connectivity (Low Priority)
- Full Portscan of Host (Low Priority)

Worm Propagation

Worm Propagation can be detected in the very early stages of the attack. As the worm is launched it will start random connection attempts to identify possibly vulnerable systems that can be infected. These connection attempts will hit many non-existent addresses, undoubtedly several of which will be reported back to the central collector of ICMP-T3 messages.

There are several key elements in worm propagation that can be seen clearly in the ICMP-T3 messages. Usually as soon as a system is compromised it will start a scan of its own to identify more vulnerable systems. This scan will look exactly like the initial scan except for the source address. Another machine is now also scanning for the same vulnerability. The pattern stays the same even if the worm is scanning for multiple vulnerabilities. This means that the packets will be directed towards a certain port (for UDP/TCP) or type/code couple (ICMP).

From a host that is performing such a scan the ICMP-T3 messages will have the following features:

- 1 Source Address
- Many Different Destination Addresses
- TCP/UDP port or ICMP type/code will be Constant

The above behavior, coming from one source address, will be referred to as a "bloom". If, over time, the number of blooms that display similar behavior increases then this means that a worm is propagating and effectively infecting systems. Note that ICMP-T3 is normally only generated for addresses that do not exist, so the actual infection can only be induced from the fact that the newly infected machine just started scanning by itself with the same properties.

Host or Network loses Connectivity

When a host or network loses connectivity it could be a sign of an attack in progress. If a DDoS Syn Flood attack succeeds and the host loses its connectivity, then the DDoS attackers will continue to send Syn packets. This is a lower priority event, which will be discarded in a shorter period of time than possible worm propagation events. The pattern would be:

- 1 Destination Address

- Many Different Source Addresses
- Several TCP/UDP ports or ICMP type/codes

Full Portscan of Host

When a host is scanned for available services ICMP-T3 port unreachable messages will be generated. The scan will be either linear or randomized and will connect to many ports on one target machine. Note that this is also a lower priority event, which will be discarded in a shorter period of time than possible worm propagation events. The pattern is:

- 1 Source Address
- 1 Destination Address
- Many Different TCP/UDP ports or ICMP type/codes

4. PRELIMINARY RESULTS

4.1. Setup

To collect the ICMP-T3 messages we construct routers with a modified version of the Linux Kernel. The modification forwards each generated ICMP-T3 message to a central collector. This collector then buffers and fowards the ICMP-T3 messages to one or more analyzers.

Due to the enormous amount of comparisons that need to be done we were forced to construct a scheme in which the data can be split and processed in paralel. First the entire IP range is split over all the available analyzers. Each analyzer will then take care of analyzing a portion of the address space.

When a packet comes into the collector, two copies are made in order to sort them by both embedded source IP address and embedded destination IP address. The packets are sorted on IP address in the analyzers. This usually means that two analyzers get a copy of the ICMP-T3 packet, only when both the embedded source address and the embedded destination address fall in the same IP range then one analyzer gets both copies. This way it can be easily determined whether one IP address contacted many different other machines, or if one IP address got contacted by many others.

Next the analyzers will attempt to locate patterns in the list of source IPs and destination IPs. If the ICMP-T3s generated by one IP address exceed a certain threshold more checks are performed. First a count is done on the transport protocols used. Then a count is done on the port numbers used for both source IP, destination IP for each source port and each destination port. (In case of the ICMP protocol the same count is done for type and code.) This count will yield information on whether one IP contacted many others on just one port, or one other machine on many ports.

If a pattern (as giving in the previous section) is detected, a packet signature is created and send to a correlator. The correlator is the final step in the process where all the reports of the analyzers come together. If an event is detected (for example a "bloom"; one machine contacting many on just one port) then the correlator compares it to the previous events. If there are sufficient similarities the events can be coupled. This could then be a sign of worm propagation.

It is obvious that the detection system contains many parameters and thresholds. As of now we have no clear idea how different values influence detection of real world scenarios. The next step in our research is to run our test cases agains different parameter sets and evaluate the performace.

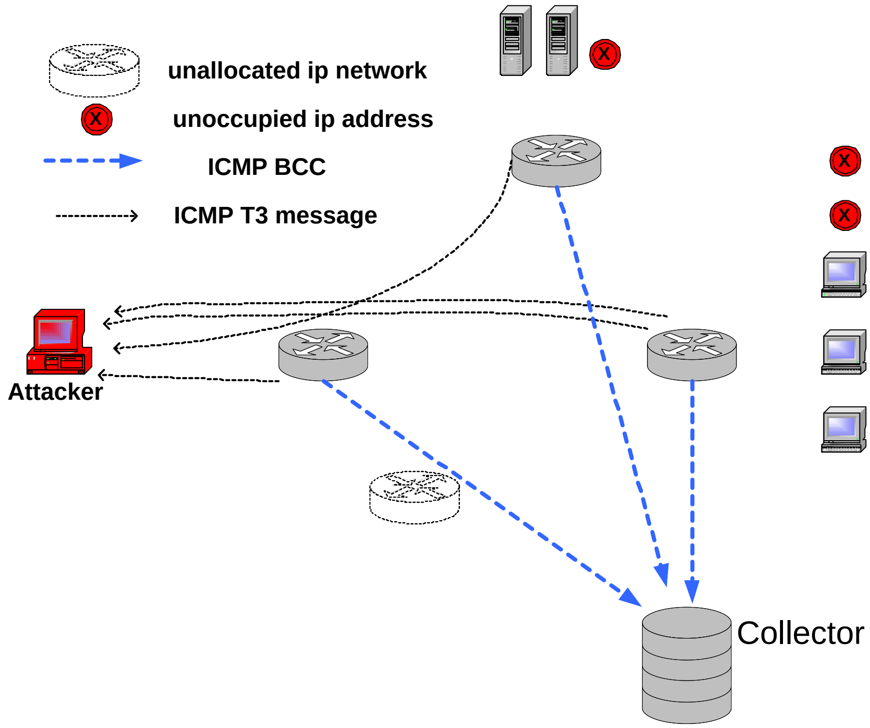

Figure 4. ICMP Destination Unreachable messages are duplicated in the router and forwarded to the central collector.

4.2. Test Data and Preliminary Results

In order to generate test data we referred to work done by Weaver (14) on worm propagation, Moore (8) on the spread of the Code-Red V2 Worm and Spafford (7) on the Morris worm. The current framework is still under development and we rely on our real world routers for more accurate test data.

By configuring a router to route to a large, non-existent network (class A size) and using a second machine to generate scans of that address space we generate the ICMP-T3 messages, which are subsequently sent to the collection point. The interaction and forwarding of ICMP-T3 messages is shown in figure 4. We use the scanning machine to scan fifty or more random IP addresses at a time. Since all these addresses are non-existent all of the accesses will trigger a ICMP-T3 response, which are all sent to the collection point for analysis.

To simulate background noise we inject a baseline (2 Mbit/second) of random ICMP-T3 messages from a dataset that was captured during normal Internet traffic. The goal is to detect the scans done by the scanning machine regardless of the rate of baseline ICMP-T3 traffic. At rates up to 30 Mbps (which is the upper limit of the test equipment) all scans could be observed.

4.3. Future Work

Most of the current parameters are set to the scenario as seen during the propagation of the Code-Red V2 worm. As we are expanding our group of participating routers we are gathering more and more real world data on which to base our parameter set. Our current research effort is aimed at aquiring more participants and finetuning the parameter sets.

Due to the enormous amount of ICMP-T3 data that is generated during actual active worm propagation our goal is to implement a pre-filter that will filter out known events. When a worm pattern has been identified there is no longer a need to collect more ICMP-T3 data triggerd by it. This way resources are kept free to identify as yet unobserved events.

For historical analysis of worms we are constructing a packet database that keeps a copy of every incoming ICMP-T3 packet, even of those that get filtered out by the pre-filter. Due to the storage requirements, this database will be a FIFO storage so that all data is continously being rotated.

5. SUMMARY

In this paper we presented a new method to detect worm propagation across the Internet. By using Blind Carbon Copies of ICMP Destination Unreachable messages generated by routers located all over the Internet we identify the scanning behavior of worms. Those ICMP-T3 are directed to a central collection point where analyzers try to identify similarities between the incoming packets.

When a worm scans for other vulnerable hosts it hits addresses that are not actually represented by a physical host. This will trigger a router to return an ICMP-T3 Destination Unreachable message. When those routers send a copy of this message to a central point of collection a clear pattern can be discerned. When a single host scans across the Internet for a certain vulnerability then this can be seen as a 'bloom' of related activity. When moments later one or more other hosts show the exact same kind of activity (in the form of blooms), scanning for the same vulnerability, then this is a clear sign of a worm propagating the Internet.

Although in early stages of development we have already achieved success in a test environment where the propagation of worms was simulated. Subsequently our research will now focus on determining the best set of parameters and aquiring more routers across the globe to get a more acurate view of real world worm propagation.

REFERENCES

1. http://www.incidents.org/react/nimda.pdf, "Nimda worm/virus report.".
2. A. Boulanger, "Catapults and grappling hooks: The tools and techniques of informatoin warfare," 1998.
3. http://www.phrack.org/phrack/49/P49 15, "Port scanning without the syn flag.".
4. T. H. Project, *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community*, Addison-Wesley, 2001.
5. J. Postel, "Rfc 793: Transmission control protocol," *Request for Comments* **793**, 1981.
6. http://www.insecure.org/nmap/nmap_doc.html, "The art of port scanning." 1997.
7. E. H. Spafford, "The internet worm incident technical report csd-tr-933," *Lecture Notes in Computer Science* **87**, Springer-Verlag, 1989.
8. D. Moore, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," 2001.
9. http://www.eeye.com/html/Research/Advisories/AL20010717.html, ".ida code red worm." 2001.
10. http://www.incidents.org/react/code_red.php, "Code red." 2001.
11. http://www.incidents.org/react/code_redII.php, "Code red 2." 2001.
12. D. C. Plummer, "Rfc 826: An ethernet address resolution protocol," *Request for Comments* **826**, 1982.
13. J. Postel, "Rfc 792: Internet control message protocol," *Request for Comments* **792**, 1981.
14. N. Weaver, "Potential strategies for high speed active worms: A worst case analysis," *Usenix Security* , 2002.
15. http://www.cisco.com/warp/public/707/GSR-unreachables pub.shtml, "Icmp unreachable vulnerability in cisco 12000 series internet router.".
16. Y. Rekhter and T. Li, "Rfc 1771: A border gateway protocol 4 (bgp-4)," *Request for Comments* **1771**, 1995.
17. G. Mansfield, K. Ohta, Y. Takei, N. Kato, and Y. Nemoto, "Towards trapping wily intruders in the large," 1999.
18. J. Postel, "Rfc 791: Internet protocol," *Request for Comments* **791**, 1981.
19. http://www.caida.org/analysis/security/code red/coderedv2_analysis.xml, "The spread of the code-red worm (crv2).".
20. http://www.silicondefense.com/cr/july.html, "Analysis of spread of july infestation of the code red worm.".