

Please type a plus sign (+) inside this box → +

PTO/SB/16 (8-00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(S)					
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)			
Chris	Jordan	Leesburg, Virginia			
☐ Additional inventors are being named on the _____ separately numbered sheets attached hereto					
TITLE OF THE INVENTION (280 characters max)					
Device, System, and Method for Defending a Computer Network					
CORRESPONDENCE ADDRESS					
Direct all correspondence to:					
☒ Customer Number		34060	→ Place Customer Number Bar Code Label here		
OR Type Customer Number here					
☐ Firm or Individual Name	Michael N. Haynes				
Address	1341 Huntersfield Close				
Address					
City	Keswick	State	VA	ZIP	22947
Country	US	Telephone	434 972 9988	Fax	413 375 4232
ENCLOSED APPLICATION PARTS (check all that apply)					
☒ Specification Number of Pages		17	☐ CD(s), Number		
☐ Drawing(s) Number of Sheets			☒ Other (specify) Certificate of Express Mailing		
☐ Application Data Sheet. See 37 CFR 1.76		Fee Transmittal			
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)					
☒ Applicant claims small entity status. See 37 CFR 1.27.					
☐ A check or money order is enclosed to cover the filing fees					
☐ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 50-2504					
☒ Payment by credit card. Form PTO-2038 is attached.					
FILING FEE AMOUNT (\$) 80.00					
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.					
☒ No.					
☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____					

Respectfully submitted,

SIGNATURE _____

TYPED or PRINTED NAME Michael N. Haynes

TELEPHONE 434 972 9988

Date

17 Nov 2003

REGISTRATION NO.

40,014

(if appropriate)

Docket Number:

1056-002

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C., 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

22582 U.S. PTO
60/520577



22783 U.S. PTO
11703

PTO/SB/17 (10-03)

Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☒ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$ 80.00

Complete if Known

Application Number	
Filing Date	17 November 2003
First Named Inventor	Chris Jordan
Examiner Name	Not yet assigned
Art Unit	Not yet assigned
Attorney Docket No.	1056-002

METHOD OF PAYMENT (check all that apply)

☐ Check ☒ Credit card ☐ Money Order ☐ Other ☐ None

☐ Deposit Account:

Deposit Account Number
Deposit Account Name

50-2504

Michael N. Haynes

The Director is authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	80.00
SUBTOTAL (1)					(\$ 80.00

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

Total Claims	Extra Claims	Fee from below	Fee Paid
	-20** =	X	
Independent Claims	-3** =	X	
Multiple Dependent			

Large Entity		Small Entity		Fee Description
Fee Code	Fee (\$)	Fee Code	Fee (\$)	
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2)

(\$)

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for <i>ex parte</i> reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify)

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$)

SUBMITTED BY

(Complete if applicable)

Name (Print/Type)	Michael N. Haynes	Registration No. (Attorney/Agent)	40,014	Telephone	434 972 9988
Signature		Date	17 Nov 2003		

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

CERTIFICATE OF EXPRESS MAILING

Express Mail Mailing Label Number ER 422197288 US

Date of Deposit 17 November 2003

Pursuant to 37 C.F.R. § 1.10, I certify that I am personally depositing the following paper(s) or fee(s) with the "Express Mail Post Office to Addressee" service of the United States Postal Service on the above date in a sealed envelope (a) having the above-numbered Express Mail label and sufficient postage affixed, and (b) addressed to: Mail Stop Provisional Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA, 22313-1450.

Stamped, self-addressed postcard for USPTO receipt stamp (1 card)

PTO/SB/16 Provisional Application for Patent Cover Sheet (1 sheet)

PTO/SB/17 Fee Transmittal Form (1 sheet)

PTO-2038 Credit Card Payment Form (1 sheet)

U.S. provisional patent application (17 sheets)

Serial No. PROVISIONAL APPLICATION
Application Filing Date: 17 November 2003
Document Submission Date: 17 November 2003

Inventors: Chris Jordan
Docket: 1056-002

17 November 2003
Date

Eden Brown
Name of Person Making Deposit

Eden Brown
Signature of Person Making Deposit

Device, System, and Method for Defending a Computer Network

Introduction

- [1] This document describes various exemplary embodiments of a Dynamic Response System (“DRS”) that can automatically and/or dynamically defend a computer network (including attached computers) from attacks by using common traits of attacks, such as vulnerability scanning. Vulnerability scanning is most frequently utilized by attacks such as worms and auto-rooters. DRS can take advantage of vacancies in the address-to-port mapping of a network to automatically determine packets containing malicious payloads of attacks. DRS can automatically update its traffic filtering capability to prevent such packets from entering or leaving (for quarantine reasons) the network.
- [2] In certain exemplary embodiments, DRS can automatically provide any of at least five capabilities:
- 1) prevention of compromise caused by new (day zero) attacks;
 - 2) quarantine of compromised systems;
 - 3) deterrence of unauthorized communications;
 - 4) providing of a dynamic honeypot without configuration or network mapping and/or
 - 5) gathering of forensic data for intrusion analysis while providing statistically significant protection of the network from compromise.

Definitions

- [3] As used herein, the following terms have the accompanying definitions:
- [4] **Check payload** - the section of a communication that is used to determine likelihood of vulnerability. Attackers often use a vulnerability scanner to look for vulnerable systems. Check payloads sometimes exist in automated attacks when the malicious payload is large, allowing the scanner to check more machines by first checking to see if the system is vulnerable before taking the time necessary to send the malicious payload.
- [5] **Malicious payload** - the section of a communication that actually exploits the network and/or system security fault. Unlike the check payload, this communication can be harmful to the security posture of the network and/or system.
- [6] **Sterilization** - a technique that filters outgoing traffic to remove covert traits, such as banner tags and application unique errors, which could otherwise allow attackers to gain information regarding a network’s assets.
- [7] **Tell** - when used as a noun, a trait that allows an attacker to determine something about a service and/or network.

- [8] **Topology masking** - the removal of system and/or network fingerprinting attributes.
- [9] **Honeypot** – a decoy service whose use signifies likely malicious or improper network activity.
- [10] **IP (Internet Protocol)** – a communicational protocol that allows certain systems to communicate to each other without necessarily having a direct connection therebetween. Most computer systems use Internet Protocol version 4. As used herein, “IP” refers to the protocol in general along with its residing protocols.
- [11] **Filtering Rule** – an association of a dataset and/or description of a network activity and an action. An action can include, for example, dropping a communication and/or a packet thereof, terminating a communication session, changing the routing of a communication and/or a packet thereof, and/or modifying a content of a communication.

Taking Advantage of Scanning

- [12] When partitioning a network’s IP address space, there are a substantial number of IP addresses and port numbers that typically are not assigned to an active service. Often it is the case that a system performs only two or three services out of the 1024 port addresses assigned to root level usage, such as name services (53), web (port 80), mail (port 25) and file transfer protocol services (port 21). This means that there are a substantial amount of unused address-to-port mappings that are unassigned for a typical network.
- [13] For an attack to occur, the attacker typically first determines the address-to-port mapping that is associated with a possible vulnerability. Such a discovery, especially when performed by automated scripts such as worms, can be accomplished by scanning an address range for a given service, or even a number of services. Because most mappings do not have an active service there are few replies.
- [14] The scanning technique is very prevalent in network attacks. For example, approximately 80% of all internet-based attacks are related to worm based using scanning as the means for target determination. Moreover, approximately 85% of all other attacks also use vulnerability scanning. The net result is that only about 3% of attacks are not directly related with scanning, or approximately 97% of all attacks are preceded by scanning activity.
- [15] In certain situations, the efficiency of a worm’s growth can make the worm vulnerable to a scan capture and filter response. An analysis of worm growth cited that the successes and growth curves of Code Red I, Code Red II, and NIMDA were directly related to the efficiency of their scanning. The analysis

outlined other growth mechanisms that allowed for quicker growth. These quicker growing worms were able to determine vulnerable systems quickly. The more efficient growth was accomplished by the most efficient scanning techniques that were extremely sequential. Even when creating "hit lists" to speed the growth, scanning was required.

- [16] In certain exemplary embodiments, DRS can take advantage of these worms' strength in order to defeat them. The Scan-and-Exploit defensive technique of DRS can be considered as three-fold: first, the attacker will likely visit a vacant address-to-port mapping before connecting to an active one. Second, if DRS can entice the attacker to attempt to exploit a fake service, DRS now will have a copy of the malicious packet that can be used to generate a filtering rule to filter future malicious packets from entering the network. But before the new filtering rule is implemented, DRS can test the filtering rule to previous traffic to insure that it does not impact operations. Finally, if all services respond as active, the attacker will have gained nothing by the scan. Without knowledge of vulnerable systems, a worm can be slowed, cannot use hit lists, and/or will need to use its malicious payload blindly, thereby allowing its malicious payload to be captured and used for a defensive filter.

DRS Structure

- [17] In certain exemplary embodiments, DRS can be viewed as three automated modules (a Traffic Filter, a Virtual Space Engine, and an Analytical Engine), with each module residing on a separate system. In certain exemplary embodiments, DRS can be implemented by integrating the three modules in a single automated system and/or device (appliance version) that allows for integration into a firewall or intrusion prevention system. These three modules can work together to determine new filtering rules that can be applied to traffic passing through the Traffic Filter.
- [18] In certain exemplary embodiments, the Analytical Engine can create filtering rules. The Analytical Engine can transfer the filtering rules to the Traffic Filter. The Traffic Filter can implement the filtering rules to automate the process and prevent future attacks.
- [19] In certain exemplary embodiments, the Traffic Filter can be positioned on the perimeter of the network. The Traffic Filter can remove determined "known bad" activity from the data stream based on the filtering rules. Normal Traffic can be recorded and sent to its proper destination. Communication requests to non-existing services can be sent to the Vacant Space Engine.
- [20] In certain exemplary embodiments, the Vacant Space Engine can continue a misdirected conversation, recording the packets so that they can be analyzed. The Vacant Space Engine can separate the likely bad elements and associate them with the packets generated during the same communication stream. In this manner, the Vacant Space Engine can take advantage of the fact that all packets

are not uniformly “bad”, but instead packets that are “bad” tend to occur in groups. The Vacant Space Engine need not necessarily entrap the attack, but can instead get the attack to transmit its check payload or malicious payload. Then, the Analytical Engine can extract these payloads for developing filtering rules. Follow-on communications initiated by the attack script can be unable to find or compromise services in the network.

- [21] In certain exemplary embodiments, the Analytical Engine can receive two sets of data. The Analytical Engine can receive the recorded streams from the Vacant Space Engine, and can treat this stream as if it contains malicious payloads, since typically there are not any approved services in the Vacant Space Engine. The packets from the Vacant Space Engine can be divided by the Analytical Engine into several groups, such as for example: total, checked, malicious, and/or good packets. The Analytical Engine can devise a filtering rule that supports this division. The Analytical Engine can evaluate the new filtering rule by comparing the filtering rule against a previous collection of allowed data.
- [22] In certain exemplary embodiments, the Analytical Engine can also receive a second stream of data comprising normal traffic that was directed against known services. The Analytical Engine can compare the filtering rules against the normal data. If there is no match, the Analytical Engine can add the filtering rule to the Traffic Filter. In this manner, future packets with the same structure will not enter the network, yet the Traffic Filter will not add a filter rule that affects normal traffic.

DRS Capabilities

- [23] In certain exemplary embodiments, DRS can include the capability to:
- handle state oriented interactive protocol (like FTP);
 - handle UDP protocols (like DNS);
 - handle generic responses (like character streams); and/or
 - mimic protocols in normal traffic streams.
- [24] In certain exemplary embodiments, DRS can be capable of handling attacks based on Internet protocols such as standard RFC protocols, generic protocols, generic data stream protocols, and/or Hypertext Transfer Protocol (HTTP) and residing applications, including applications for Unix, Linux, Windows, Macintosh, Palm, and/or WinCE, etc., operating systems. For example, DRS can be capable of handling attacks on Web Services, which is a commonly attacked collection of HTTP-based applications that is of high concern for many organizations.
- [25] In certain exemplary embodiments, DRS can be capable of handling Internet based services such as SSH (a secure remote login), FTP (file transfer), SMTP (mail), IMAP (mail), and/or POP3 (mail).

- [26] In certain exemplary embodiments, DRS can be capable of handling UDP services, such as DNS. UDP services can involve non-connection oriented packets. UDP protocols can be timing oriented and DRS need not fake a response while waiting for confirmation whether the service is active. DRS can also develop a generic response capability that it can use for any service port that it does not have a known response for. DRS can provide a response for all port requests so that for a process scanning from outside the network, all services appear to be open.
- [27] In certain exemplary embodiments, DRS can include a capability for the Vacant Space Engine to use normal traffic to determine how an unknown port should respond. In this manner the system can give a response to a protocol that is not already preprogrammed for and have a similar response that could trigger the malicious payload needed to protect the actual service. The difficulty in this cycle is not being able to determine further conversation packets from normal streams, but to do so without releasing sensitive information.
- [28] In certain exemplary embodiments, the Vacant Space Engine can maintain a communication long enough to capture either the check payload or the malicious payload. In certain exemplary embodiments, the Analytical Engine can produce filtering rules that are specific enough to stop malicious packets while not impacting normal traffic.
- [29] In certain exemplary embodiments, to increase the ease of implementing DRS, there are a number of unique concepts that can be used:
- 1) Sanitization;
 - 2) Just-In-Time (JIT) Discovery; and/or
 - 3) Topology Masking.
- [30] In certain exemplary embodiments, DRS can remove malicious payloads from reaching active systems. DRS can enhance the likelihood of determining a filtering rule for malicious behavior by using network sessions that have a statistically high likelihood of being malicious (such as sessions to non-existing services). DRS can test these new filtering rules to ensure that they do not impact operations (such testing can be done by limiting the filtering rule generation capability) before implementing them.
- [31] In certain exemplary embodiments, via topology masking, DRS can defeat "scanning" by providing little or no knowledge of the protected network and/or network service. An attacker can be forced to launch the attack against the network service on its default port and against every address.

- [32] In certain exemplary embodiments, the Analytical Engine can capture the payloads from attacking scripts and filter out those payloads from incoming traffic. These filtering rules can be passed onto the Traffic Filter for implementation, which can prevent future compromises (prevention).
- [33] In certain exemplary embodiments, the Traffic Filter can contain a detected attack by using the identified malicious and check packets preventing them from leaving the network (quarantining).

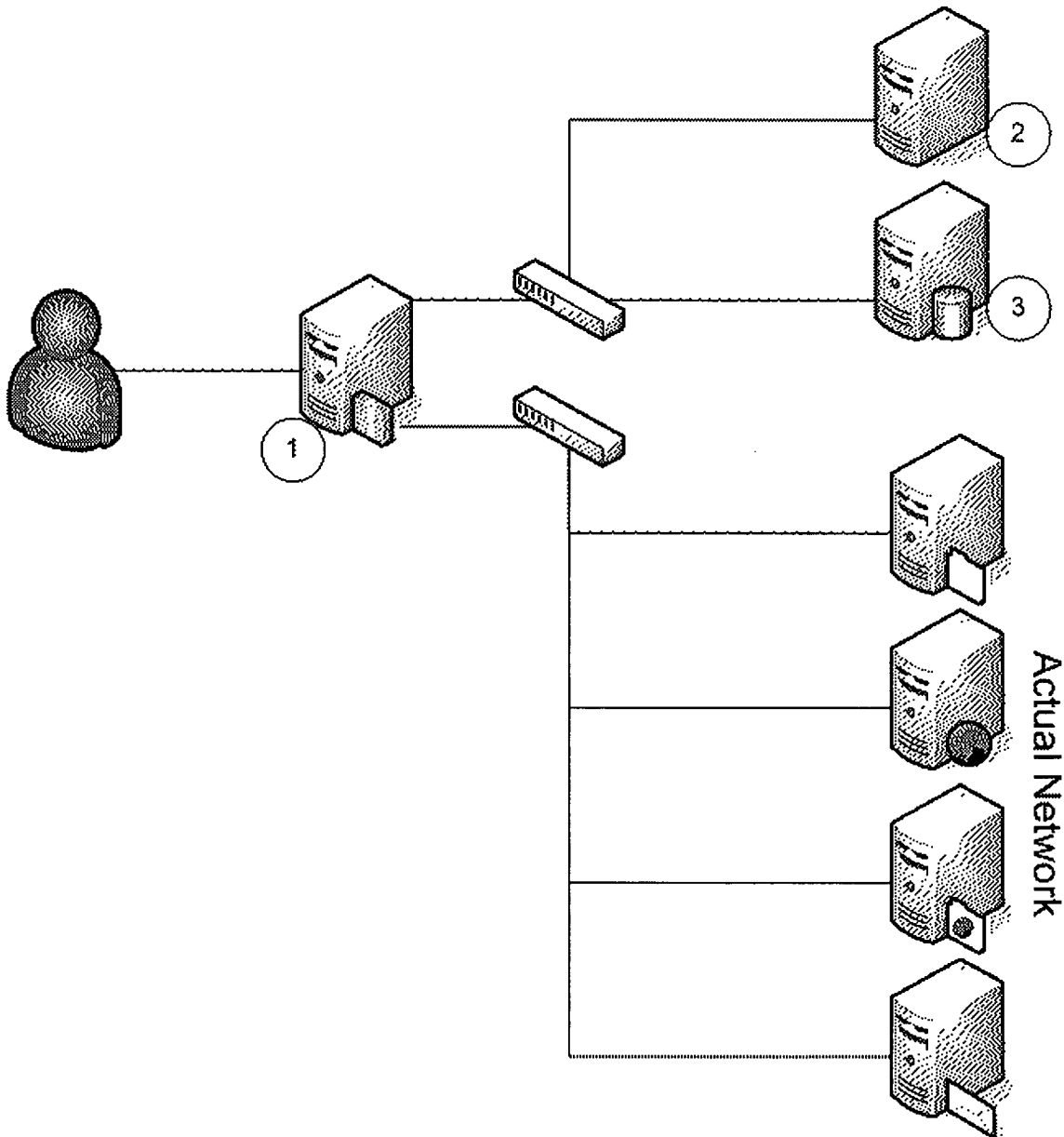


Figure 1 - General Overview of Potential DRS Functions

- [34] **FIG. 1** is a block diagram of an exemplary embodiment of a DRS, showing potential functions of potential components. In certain exemplary embodiments, the following DRS components can provide the listed functions:
- [35] The Traffic Filter (labeled 1 in FIG 1.)
- Routing to non-existing services to the virtual space engine (JIT Discovery)
 - Sanitization of sessions (both protocol and application layers)
 - Topology Masking.
 - Filtering of known malicious traffic as defined by the Analytical Engine and pre-defined filtering rules.
- [36] The Virtual Space Engine (enhanced shallow honey pot, labeled 2 in FIG 1)
- Interaction to sessions
 - Recording of communications to virtual space engine
- [37] The Analytical Engine (labeled 3 in FIG 1)
- Receives information from both real and decoy systems
 - Learning Engine
 - Generates both trivial and learned filtering rules
 - Test filtering rules to determine operational impact (effectiveness)
- [38] In certain exemplary embodiments, DRS can dynamically respond to new attacks using the filtering rules created based on the anomalous connection to non-existing services. In certain exemplary embodiments, DRS can automatically detect, analyze, develop filtering rules, and/or modify the security posture of the network. This can be done by using high likelihood malicious anomalies, such as the anomaly caused by connection attempts to non-existent services. In this case, DSR can communicate to the connecting session request when the actual service does not exist, and DSR can records and analyze the communication session. DRS can develop filtering rules based on the recorded malicious payloads in near real-time. DRS does not necessarily need to be configured with the topology of the network.

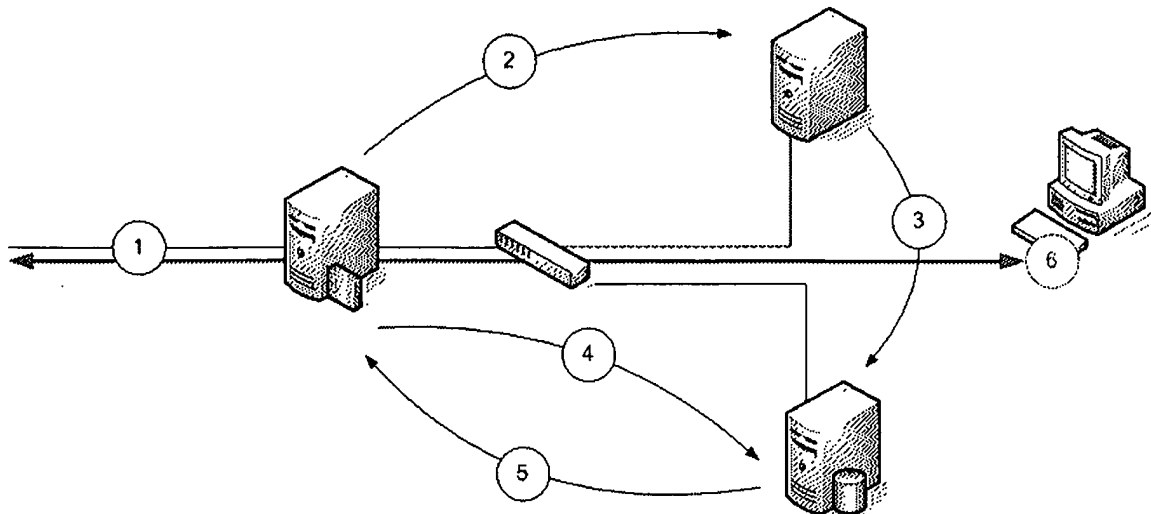


Figure 2 - Flow of a System-to-Component DRS

- [39] **FIG. 2** is a block diagram of an exemplary embodiment of DRS, showing three modules of DRS (namely the Virtual Space Engine, the Analytical Engine, and the Traffic Filter). FIG. 2 outlines a possible flow of interaction:
- 1) A session comes from outside the perimeter.
 - 2) If the session attempts to connect to a service that is not running, then the Traffic Filter sends the session request and further associated packets to the Virtual Space Engine.
 - 3) Upon recording an anomalous session, the Virtual Space Engine sends the session data to the Analytical Engine.
 - 4) If a session was to a legitimate service, the Traffic Filter records the conversation and sends the resulting communication to the Analytical Engine. The Analytical Engine now compares the traffic received by the Virtual Space Engine to that of the Traffic Filter. If there is a means to describe the traffic from the Virtual Space Engine without impact to previous "good" data, then (goto 5)
 - 5) the Analytical Engine now sends a filtering rule to the Traffic Filter. If the Analytical Engine derives a filtering rule that is the composite of other filtering rules, it sends a list of what filtering rules are no longer needed.
 - 6) Normal traffic passes through the Traffic Filter, where malicious packets are dropped according to the filtering rules.
- [40] In certain exemplary embodiments, to eliminate the need of mimicking several versions of the same protocol during communication, a second technology which we call "Sterilization" can be used by DRS. The technology can not only simplify DRS, but also can conceal internal system information from outside attackers.
- [41] In certain exemplary embodiments, by using a known negative trait, scanning, DRS can use that to extract further malicious data to create an automated response. Scanning is just one malicious trait. By examining traffic flow and

determining other traits, further automated responses can be developed based on the same process of automated extraction, testing, and/or implementation.

Traffic Flow Control

- [42] In certain exemplary embodiments, DRS can be divided into three modules where each module resides on a separate system (system-to-component approach, see Figure 2 - Flow of a System-to-Component DRS). In certain exemplary embodiments, DRS can integrate the three components in a single device (appliance version) that can allow for integration into a firewall or intrusion prevention system. DRS can control the flow of packets (via the Traffic Filter), determining what packets are dropped, redirected, and/or mangled.
- [43] Though it can be easier to think of DRS as an interceptor that resides on the perimeter of a network, DRS does not necessarily need to reside in that position. As long as DRS can intercept and influence the flow of communications it can be positioned anywhere in the network configuration. For example, DRS can perform its role by intercepting discovery packets like Address Request Protocol (ARP) requests and redirect the flow of traffic when the request is not responded to in a timely manner.
- [44] The Traffic Filter can remove determined “known bad” activity from the data stream based on the filtering rules. These filtering rules can be created by the Analytical Engine. DRS can record normal traffic and sends that traffic to its proper destination. Communication requests to non-existing services can be sent to the Vacant Space Engine.

Just-In-Time Discovery

- [45] In certain exemplary embodiments, DRS can use passive and/or active mapping algorithms to determine if a service is active in a network. In certain exemplary embodiments, DRS can use a just-in-time (“JIT”) discovery concept. When implanting the JIT discovery approach, DRS can use the initial SYN to determine if the service is active in the network. The three-way handshake can be independent of the initiating systems handshake to the Traffic Filter. The difference between the initial handshake parameters can be mangled by the Traffic Filter to insure that communications is maintained smoothly.
- [46] **FIG. 3** is a block diagram that illustrates an exemplary embodiment of JIT Discovery.

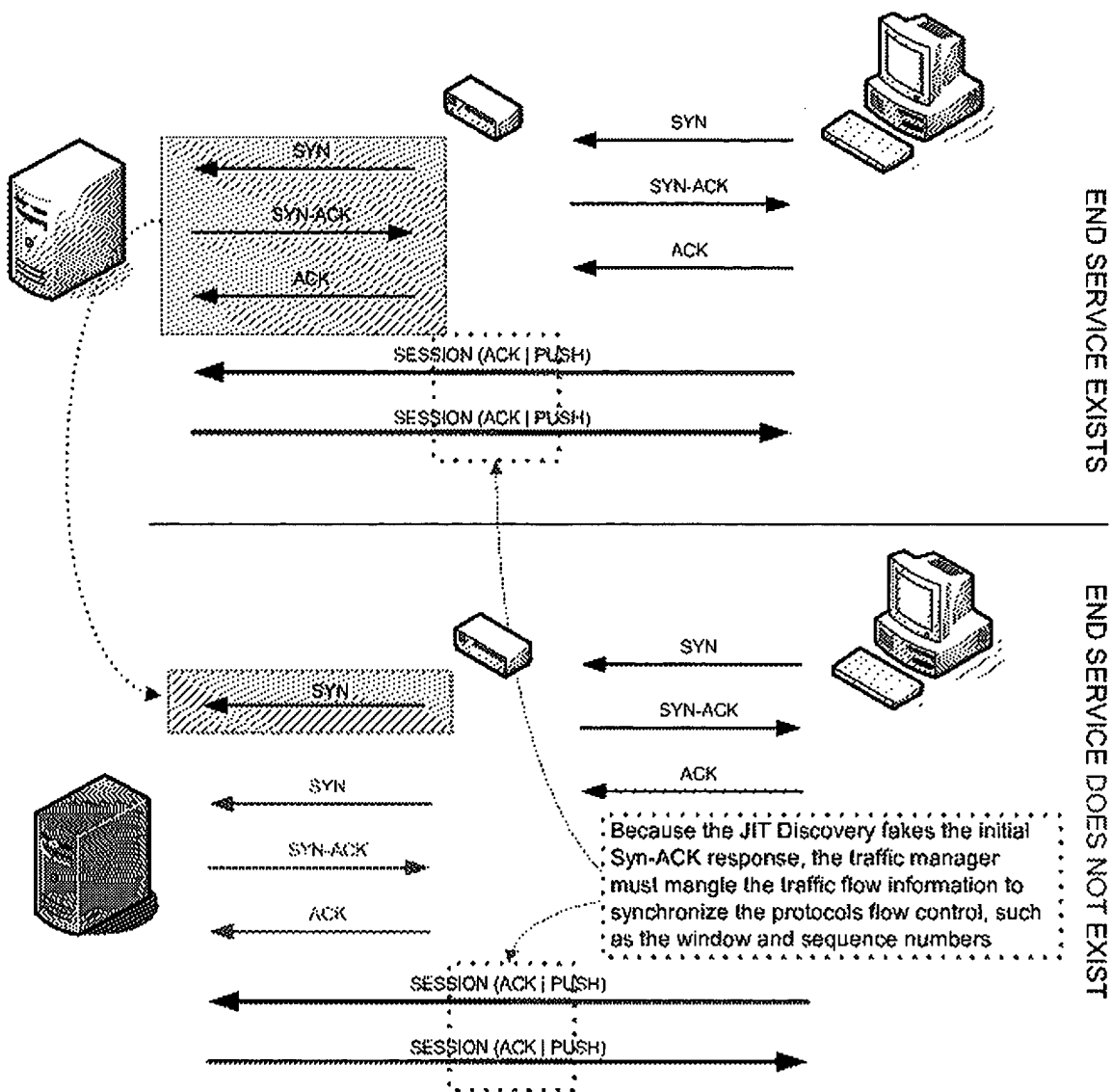


Figure 3 - Just in Time Discovery

Sanitization

- [47] In certain exemplary embodiments, DRS can sanitize communications that cross a network boundary to prevent network knowledge from being gained. By removing network traits from responses (see Figure 5 - Topology Masking), attackers can be prevented from identifying a live system from a fake system, and can be prevented from determining the application type or version. Via sanitizing, any or all services and/or systems can appear to the attacker to be the same and therefore it is likely that only one version of the communication protocol needs to be mimicked by the Virtual Space Engine (there are a few protocols like secure shell where the header can be required to determine communication type).

- [48] DRS sanitization can replace a “tell” with an alternative that is consistent with any other communication leaving the network. For example, Microsoft’s IIS has error codes like “404.1 – Site not found”. DRS sanitization can replace that response with the more generic “404 Not Found”. Thus, an attacker can lose the ability to determine if the web server is a possible vulnerable IIS server if there are no tells in the communication.
- [49] DRS sanitization also can make it so the vacant space engine does not need to mimic a number of different versions of services. When there is a heterogeneous network, a honey pot can have a lower chance to catch an attack before the attack hits an actual service. DRS sanitization can address this problem.
- [50] **FIG. 4** illustrates an exemplary calculation for decreasing the odds of attacking a real system first.

$$\text{Probability of Exploiting the Target} = \frac{\text{Number of Actual Target}}{\text{Total Number of Actual}}$$

Consider, the target image is a Windows IIS 4.0 and there are ten actual such systems in the network and fifteen systems that are Linux running Apache. If in the fake part of the network there are a proportionately the same number of server types, then in a subset of 130 images there would be 40 fake IIS images. Therefore the chance of an attacker hitting a decoy IIS server before a real one if the attacker first checked the image for possible vulnerability is:

$$20\% = \frac{10}{50}$$

Consider, the target image is a Windows IIS 4.0 and there are ten actual such systems in the network. If the remaining fake images in a subnet of 130 images are also Windows IIS 4.0 then the chance of an attacker hitting a decoy IIS server before a real one if the attacker first checked the image for possible vulnerability is:

$$7.7\% = \frac{10}{130}$$

Figure 4 - Decreasing the odds of attacking a real system first

- [51] As a secondary effect, attackers can be prevented from gaining information with their attacks. Attackers wishing to continue their attacks against the system can be forced to do so blindly.

Topology Masking

- [52] Topology masking can limit the amount of knowledge that an outsider can gain from interacting with the network. Part of this can be achieved via JIT Discovery that can make all services appear active even if they are not, which can prevent fingerprinting of the network topology. Sanitization can prevent the fingerprinting of the applications. The combination of JIT Discovery and Sanitization can address and/or prevent application, system, and/or network fingerprinting, which can thereby mask the network and/or its topology.
- [53] **FIG. 5** illustrates an exemplary relationship between actual and faked attributes, and actual and faked contents.

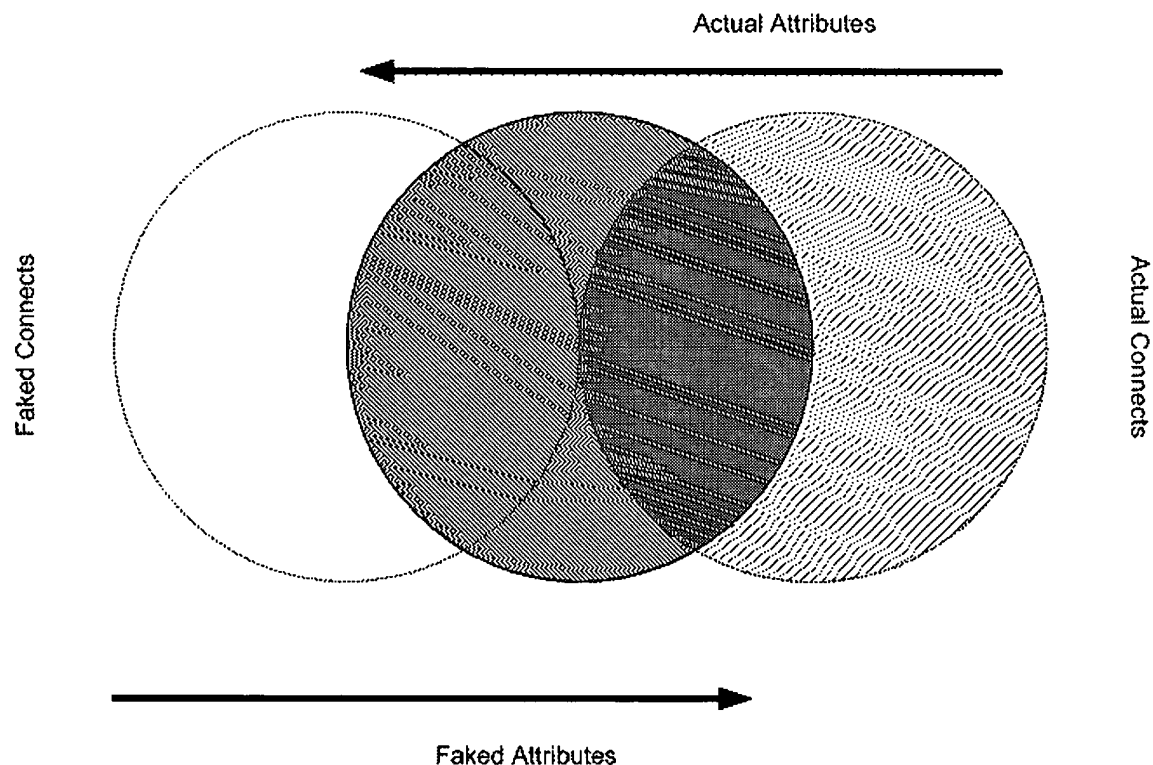


Figure 5 - Topology Masking

- [54] The success of topology masking can be related to the ability to make the attributes of a decoy system as similar as possible to those of an actual system. In certain exemplary embodiments, DRS can implement topology masking using the common technique of making decoy systems appear as actual systems. In certain exemplary embodiments, DRS also can use sanitization to make the actual systems appear more like decoy systems (see **Figure 5 - Topology Masking**). Such sanitization can hide the attributes of the entire protocol stack.

[55] **FIG. 6** describes how the benefits of scanning can be removed and/or defeated.

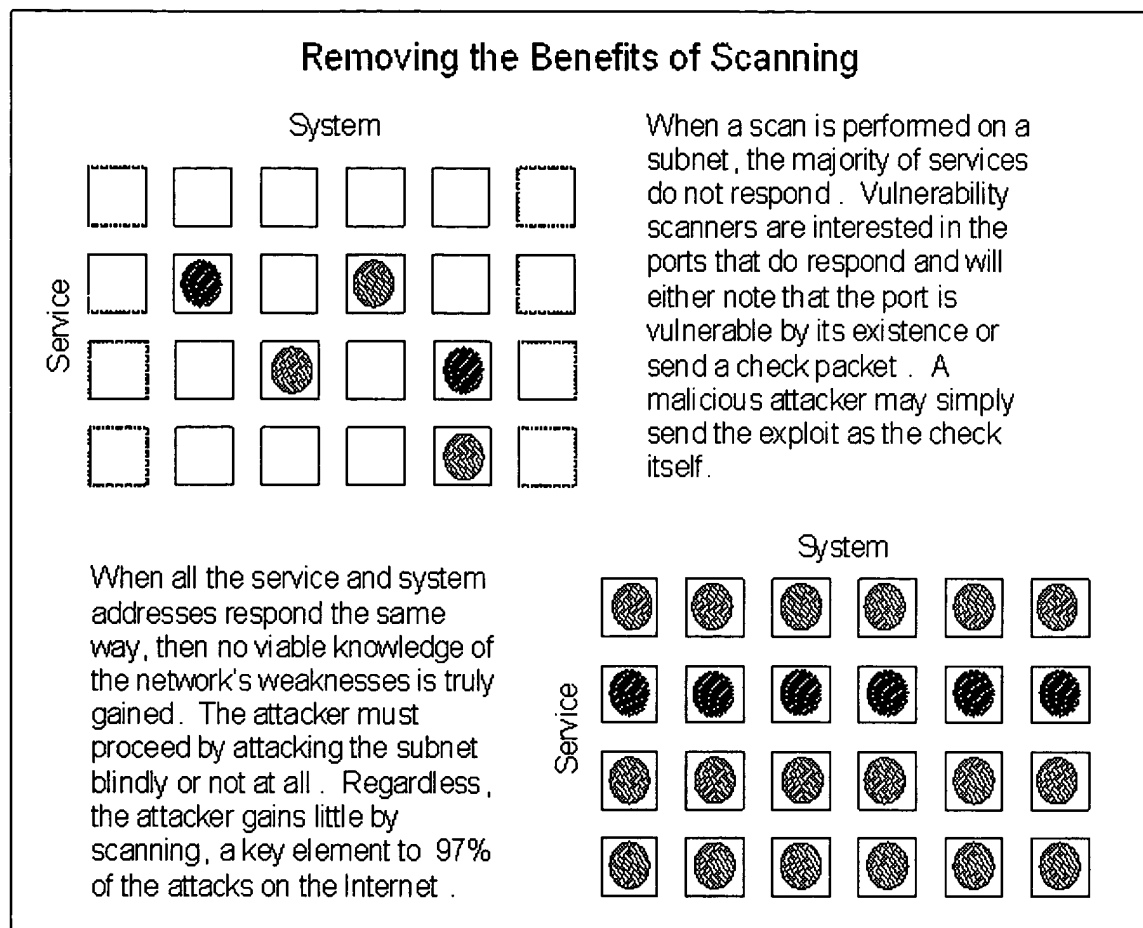


Figure 6 - Scan Defense

- [56] In certain exemplary embodiments, DRS can deliver a capability that addresses the scan & exploit technique, most noted in worms, but which can also exist in many other malicious tools such as auto-rooters. In certain exemplary embodiments, DRS can address the algorithms used for rapid growth and/or can exploit the weakness of the efficiency of these algorithms. That is, as exploitations become more efficient, they become more susceptible to automated counter-responses. The ratio of nonexistent-to-existent services will likely increase greatly as the Internet transitions from IP version 4 to IP version 6. Exploitations are likely to become even more efficient with the change in address schema between the IP protocols.
- [57] In certain exemplary embodiments, DRS can intercept incoming packets for analysis. In certain exemplary embodiments, DRS can modify any or all outgoing traffic to appear as if all the systems actually exist, all operating systems appear as being the same, all services exist (see JIT Discovery), and/or all services are the same (see Sanitization). In certain exemplary embodiments, DRS can collect traffic to the unassigned addresses and/or services and forward that traffic to an

Analytical Engine component that can automatically analyze the payloads, develop a filtering rule, and/or modify the security posture of the perimeter and/or network in order to quickly respond to the events.

Filtering

- [58] In certain exemplary embodiments, DRS can use the scanner's malicious payload against itself. Like a snake antidote, DRS can steal the venom from the attack and uses that to build a defensive filtering rule before the attack strikes a vulnerable system. DRS and/or its Traffic Filter element can be positioned on a network perimeter and/or can reply to all service/communication requests even if the requested services and/or communicators are not active on the network. The service/communication requests can be redirected to the Vacant Space Engine component of the DRS based on service availability.
- [59] In certain exemplary embodiments, the Traffic Filter element of DRS can control the speed and/or pace of network activity. As bad connections are attempted, the Traffic Filter can slow the source as it attempts to interact with services by giving the connections a higher level of scrutiny or even directing the source's attempts to a jail system. This slower pace can allow the Analytical Engine to implement filtering rules after the first attempted connection. This approach can lower the probability of the attacker sending a malicious packet to an actual system before the filtering rule can be written and implemented.

Virtual Space Engine

- [60] In certain exemplary embodiments, the DRS Vacant Space Engine can continue a misdirected conversation, recording the packets so that they can be analyzed. The Vacant Space Engine can separate the likely bad elements and/or can associate them with the packets generated during the same communication stream. In this manner, the algorithm can take advantage of the fact that all packets are not uniformly "bad", but instead packets that are "bad" tend to occur in groups. The Vacant Space Engine does not necessarily entrap the attack, but can convince the attack to transmit its check payload or malicious payload. Then, the Analytical Engine can extract these payloads for filtering rules. Follow-on communications initiated by the attack script can be unable to find or compromise services in the network without malicious and check payloads reaching the vulnerable systems.
- [61] In certain exemplary embodiments, along with JIT Discovery, an enhanced shallow honey pot can further close the gap between live systems and decoys. The enhanced honey pot can allow for multiple responses after the original exchange, so as to record possible malicious payloads that occur after the initial exchange for exploitation or discovery.

Maintain Interaction

- [62] In certain exemplary embodiments, to enable DRS to make more accurate signatures, the Vacant Space Engine can keep the attacking sessions connected as it attempts to communicate. Although not essential, DRS can implement an

enhanced shallow honey pot that can allow for the mimicking of communicational protocols in a static response sequence and/or in a dynamic response sequence. The type of honey pot is not essential, but honey pot designs that get larger amounts of interactive communications can be preferred.

Recording of Session

- [63] In certain exemplary embodiments, any or all session information can be recorded by the Virtual Space Engine. For example, the Virtual Space Engine can record incoming packets. As another example, the Virtual Space Engine can record response data to aid the Learning Engine in better understanding what filtering rule to apply.

Analytical Engine

- [64] In certain exemplary embodiments, the Analytical Engine can receive two sets of data. The Analytical Engine can receive the recorded streams from the Vacant Space Engine that treats a stream as if the stream contains check or malicious payloads. The Analytical Engine can review incoming packets received and/or can create a filtering rule to describe each packet. The second stream of data the Analytical Engine receives can be normal traffic that was directed against known services. The Analytical Engine can compare the packet filtering rules against the normal data. If there is no match, the Analytical Engine can add the filtering rule to the Traffic Filter. In this manner, future packets with the same structure can be prevented from entering the network, and the Traffic Filter does not add a filtering rule that affects normal traffic.
- [65] In certain exemplary embodiments, the Analytical Engine can receive communicational datasets from both communications going to active services (via the Traffic Filter) and that from decoy services. In certain exemplary embodiments, the former is considered statistically good while the latter is considered statistically bad.

Learning Engine

- [66] In certain exemplary embodiments, the Learning Engine can improve the accuracy of filtering rules by using a sliding window on the good dataset in order to compensate for new attacks hitting an actual service first. This good dataset can be used to validate that the filtering rules generated by the bad dataset do not impact the future good communication. To increase speed of filtering rule generation, positive filtering rules can be generated to clean known good attributes from statistically bad sets before filtering rule generation.
- [67] The term “rule” is being used, not because the Learning Engine has to be rule based, but that the communication from the Analytical Engine to the Traffic Filter is considered to be a rule or sometimes referred to as a “signature”.

Testing of Filtering Rules

- [68] Generally, for operational environments, it is considered a best practice if traffic filtering rules do not impact operational services, especially those that can impact life. Thus, before filtering rules are implemented in an operational environment, testing of those filtering rules is typically performed.
- [69] In certain exemplary embodiments, DRS can test filtering rules using the known good dataset. If a filtering rule is determined unacceptable, DRS can defer and/or sideline its implementation and/or inform the Learning Engine that the filtering rule was not used.
- [70] In certain exemplary embodiments, if a filtering rule is accepted, the new filtering rule can be sent to the Traffic Filter to be implemented. The new filtering rule can be associated with filtering rules that it now supersedes. By informing the Traffic Filter of superseded filtering rules, the Traffic Filter can keep a minimal number of filtering rules that need to be applied.

DRS' Working Together

- [71] The Traffic Filter need not perform like a "fortress with guards". The Traffic Filter can perform like a traffic cop controlling the information flow. If a filtering rule can be generated by a single DRS, then that DRS can share its filtering rules with another DRS. A receiving DRS can test a received filtering rule on its data streams for prevention and quarantine. At a higher-level, such sharing and/or integration of filtering rules among multiple DRS' can provide a basis of a survivable enterprise that is able to stop infection and contain it for recovery.
- [72] Still other embodiments will become readily apparent to those skilled in this art from reading the above-recited detailed description and drawings of certain exemplary embodiments. It should be understood that numerous variations, modifications, and additional embodiments are possible, and accordingly, all such variations, modifications, and embodiments are to be regarded as being within the spirit and scope of this description and any future claims. For example, regardless of the content of any portion (e.g., title, field, background, summary, abstract, drawing figure, etc.) of this application, unless clearly specified to the contrary, there is no requirement for the inclusion in any claim of any particular described or illustrated activity or element, any particular sequence of such activities, or any particular interrelationship of such elements. Moreover, any activity can be repeated, any activity can be performed by multiple entities, and/or any element can be duplicated. Further, any activity or element can be excluded, the sequence of activities can vary, and/or the interrelationship of elements can vary. Accordingly, the descriptions and drawings are to be regarded as illustrative in nature, and not as restrictive. Moreover, when any number or range is described herein, unless clearly stated otherwise, that number or range is approximate. When any range is described herein, unless clearly stated otherwise, that range includes all values therein and all subranges therein. Any information in any

material (e.g., a United States patent, United States patent application, book, article, etc.) that has been incorporated by reference herein, is only incorporated by reference to the extent that no conflict exists between such information and the other statements and drawings set forth herein. In the event of such conflict, including a conflict that would render a claim invalid, then any such conflicting information in such incorporated by reference material is specifically not incorporated by reference herein.

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

11/20/2003 KBETEMR1 00000095 60520577

01 FC:2005

80.00 OP

PTO-1556
(5/87)