VDS: Malware Detection System

Wu Bing †, Yun Xiaochun †
{wubing,yxc}@hit.edu.cn

Xiao Xinguang
Antiy Labs
seak@antiy.net

ABSTRACT

We present the dynamic information security theory model: P2DR to deal with malware epidemics. The model includes the following stages: detection of a malware epidemic by detection system, establishment of countermeasure strategy in strategy coordination center, activation of protection and response system, defense system activation for real-time protection and response. The process is a periodic one in which the strategy is adjusted dynamically.

In the model we propose, the detection system is considered to be the key component. Based on our analysis of traditional IDS architecture, we think that it is not suitable for inspecting high speed network traffic and monitoring multivariant malware epidemics. Therefore, we propose a parallel detection model which can be applied to the backbone network matched with appropriate protocol parsing. Normalized taxonomy is also presented for the detection rules of malware. Five detection rule sets are covered, namely, match rules based on deep content pre-processing, special algorithms, binary level, binary level requiring network information checks, and pure network information. A Virus Detection System is realized based on the framework above.

Categories and Subject Descriptors

C.2 [Computer and Communication Networks]: Security and Protection – Worms; C.2.3 [Network Operations]: Network monitoring – Malware Detection; C.2.5 [Local and Wide-Area Networks]: Internet;

General Terms

Algorithms, Security

Keywords

Network Security, Normalization, P2DR Model, Virus Detection System.

1. INTRODUCTION

Nowadays, the internet is facing more and more severe security challenges. Since 2000, many serious outbreaks of self-propagating malicious code have occurred, including large-scale outbreaks such as CodeRed, Nimda, Blaster, and Sasser and so on, causing serious worldwide damages. The speed of computer infection is getting faster and faster and doing more and more damage.

Since May 2004, popular worms have been transmitted mostly through emails with a notable increase of worms carried by instant messaging tools. Instances of malicious programs such as Troy and Backdoor also grow rapidly. Since the internet is an open, complex and vast system, security breaches are inevitable. Malware propagation events caused by self-propagating worms are the most serious network security events. Generally, network security events are dealt with using the P2DR model, in which the detection system is the basis of the model. The reason is that only when malware epidemics are detected in time and accurately can a valid protection response be executed.

In the ongoing study we are reporting, malware events are dealt with as standard network security events, in response to which a P2DR model coordinated by a strategy center is constructed. Defense work is coordinated by the strategy coordination center which has an emergency response and protection system ready to act on malware events upon detection of crisis.

In this paper, the detection procedure of the P2DR model is discussed, and its weaknesses are also given through an analysis of traditional IDS. A Malware detection architecture based on a large characteristic database, which is independent of protocol, is presented. According to different detection methods used, the existing malware is classified into basic malware, E-mail malware, URL malware and Script malware; characteristic database associations are established accordingly. Four kinds of engines are designed to detect malware so that defense strategies can be planed.

The concept of a "Virus" in the Virus Detection System mentioned in this paper is a generalized virus, which is the same as malware, including traditional PE viruses, Worms (worms based on files or Internet worms), Trojans, Backdoors, URL malware, Script malware and so on. For the continuity of the Anti-Virus concept, all malware programs are called VIRUSES in this paper.

2. ONGOING RESEARCH IN MALWARE DEFENSE

For the past decade, research on large-scale malicious code epidemics and malware countermeasures has been an ongoing process, the importance of which has risen to the level of counterattacking the information-based global security threat. The field of malware countermeasure development is active, with several new detection methods being proposed annually.

GrIDS[1], presented by Cheung et.al., is designed to analyze large-scale malicious attacks on computer networks and automatic intrusion. The system collects data of computer activity and network traffic between computers. Driven by a predefined mode

† This publication was sponsored by the National Natural Science Foundation of China under NSFC Contract 60403033.

WORM'05, November 11, 2005, Washington, DC, USA.

base, it aggregates this information into activity graphs which reveal the causal structure of network activity, matched with pre-defined behavior mode graphs so that worms will be detected if they exist. However, since it is designed for large-scale network intrusion, correlation is not analyzed between a sequence of packets without sufficient and effective data utilization. In addition, only simple relevance analysis is conducted based on events, with availability of the target address and target service unanalyzed in the TCP connection, which is important data for detecting an epidemic of network worms.

An anti-internet-worm and virus protection system with PLDs (programmable logic devices) [2] designed by John W. Lockwood is composed of 3 inter-connected parts: DED (data enabling device), CMS (content matching server) and RTP (regional transaction processor). DED captures packages, scans the datagram, matches characteristic strings or rule formulas provided by CMS and sends the result to RTP. The existing worm characteristic is read by CMS from a MYSQL database in the background, while characteristic strings and rule formulas that can be used in DED are compiled and integrated. RTP can then take actions according to the result matched by DED. When an internet worm breaks out, its characteristic will be added to the CMS characteristic database by administrators so that only the target characteristic is scanned by DED, and request of blocking or passing will be made to RTP. The deficiency of the system is that it can only work after the capture of characteristics, and is unable to detect and defend against unknown worms. In addition, the characteristics matching technology used in the system have some defects and may cause false alarm.

2.1 P2DR Model

The TCSEC (Trusted Network Interpretation of the Trusted Computer Security Evaluation Criteria, NCSC TC-005) model [3] is the representation of static computer safety models which was put forward by United States Department of Defense (NCSC) in 1985. Based on this model, Zeng Zhifeng et.al presented the P2DR model [4], which has become the primary model for adaptive network security theory (also called dynamic information security theory). The P2DR model is a development of TCSEC model, which is a widely adopted safety model. The P2DR model contains four main parts: Policy, Protection, Detection and Response. Protection, Detection and Response constitute a complete dynamic safety loop, which assures the safety of information systems under a security strategy co-ordination system. The P2DR model is shown in Fig.1

Since security events such as large-scale malware epidemics, system intrusions and so on are all standard network security events, the P2DR model is proposed to deal with security events such as malware epidemics. The procedure is as follows: according to the daily detection log, the strategy coordination center generates protection strategies, which are then sent out to network management organizations or network management devices directly through a given channel. When an epidemic breaks out on the network, worm samples are captured by a detection system with the ability to find as-yet-unknown worms. Characteristics of the worm are extracted according to definite flow and supplied to the detection system, at the same time, the strategy coordination center sends out a protection and emergency response strategy. Following the strategies, the emergency response system can deal with the epidemic. Following the real-time monitoring of the epidemic's mutation by the detection system, the strategy coordination center updates the protection and response strategy dynamically, schedules a protection and emergency response system, and supervises the entire process for the control of the epidemic.

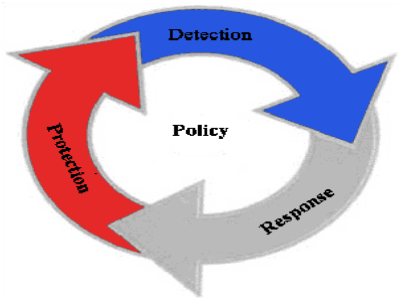

Figure 1. P^{2}DR Model.

2.2 Detection System

When handling a worm epidemic based on the P2DR model, the most important thing is to find and capture new malware samples [5] in time. Therefore, we designed two kinds of detection systems. One is VDS which inspects according to characteristics extracted from known malware samples, and the other is a malware searching and forewarning system which contains VDS but with the additional capability of finding unknown worms plus a honey pot, sample exchange system and so on. Since VDS is a line-speed detection system based on a network switch, the system only works at package or flow level and cannot recover samples detected entirely, so the entire sample cannot be obtained. To solve this problem, we designed VDS with the ability to capture unknown worms. The main difference is that it is able to realize entire flow decoding, extraction decoding, filtering at the flow level or file level and queue handling. The design of VDS will be discussed in detail in the following section. The sample sources of the detection system up to October 2004 are shown in Table1. Note that samples given by VDS or the network collection system occupy 0.51% of the total amount of samples, but samples such as Dvldr, Blaster, Welchian, and Sasser were captured in China first thanks to VDS warning.

Table 1. Malware sample source statistical table

Sample Source	Percent
Sample exchange	52.89%
User submissions	22.78%
Collecting in field	1.58%
Collecting actively	16.82%
Collection by VDS	0.51%
Collection by Honey Pots	4.44%
Other	0.98%

2.3 Emergency response system

The emergency response system is the countermeasure for known epidemics. In recent years, achievements in malicious code epidemic countermeasures have increased greatly. These achievements can be divided into three groups: quarantine defense, activity defense and worm defense.

With respect to Quarantine Defense, David Moore [5] et.al. made a study on valid quarantine for worms at the Internet level, and discussed three characteristics of a quarantine system: reaction,

containment strategy and blocking location. Three factors for improving the availability of containment are presented, namely fast, automatic detection and response to worm epidemics, content filtering and ways to cover the entire Internet when quarantine systems are adopted. Cynthia Wong [6] et.al. pointed out that limiting the contact rate at the backbone routers is substantially more effective—it renders a slowdown comparable to deploying rate limiting filters at every covered individual host. Phillip Porras [7] et.al. examined the complementary nature of two cyber defense strategies: Connection Rate Limiting representing an RL solution, and the Friends protocol, representing an LA solution. They also proposed a hybrid combination strategy that merges the two complementary quarantine techniques.

The plan behind activity defense is to establish several high-risk vulnerability databases for the OS and a relevant database of vulnerabilities that are used by the worm, and then to find worm infected computers through the detection system. After that, use a special releaser to release removal tools onto target computers, targeting the specific worm according to the established worm backdoor database and exploit database of the OS, so as to actively remove the worm. We have practiced this method and obtained fairly satisfactory results, but the validity of defense behavior still needs further discussion.

For worm defense, the key to controlling propagation is finding ways to manage and maintain those disordered and uncontrollable network nodes. To combat at malignant worm, a benign worm can be designed, which has the privilege to increase with speed greater than or equal to the malignant worm. The trace of the malicious worm's propagation can then be detected and the worm removed from the infected host. The distributed propagation technology of the benign worm is adapted to maintain and repair network nodes in time. The key issue in designing a good worm is controllability, so further factors probably need to be considered. Frank Castaneda [8] et.al. have proposed an anti-worm generation framework. They did preliminary studies on automatically transforming a malicious worm into a benign worm, on generating anti-worm code, and on the effect of four different anti-worm propagation schemes through simulation.

2.4 Strategy Coordination Center and protection Strategy

The task of the strategy coordination center is to send out strategies, coordinate the protection system and the emergency response system, and defend against the malware epidemic according to the protection and defense strategies that it devises. Router special ports or special strings are filtered by protection strategies according to the epidemic profile. Defense strategies are sent out in two ways: one is by a network security administration organization such as CERT, which is generally used to send out a defense strategy among different networks, and the other is by a routing protocol like OSPF, which is mainly sent out in a controlled network, the distributed contents are mostly protection strategies.

The major task of a strategy coordination center is to establish a countermeasure knowledge base, in which the worm behaviors database [9], and various propagation models are contained [10-14]. The strategy coordination center devises a defense strategy according to the database of worm behaviors and the propagation model, and tracks the current malware epidemic to adjust its countermeasure strategy in time.

3. FROM IDS TO VDS

3.1 Architecture Limitations of Traditional IDS

At present, the traditional network IDS adopts a framework based on detailed protocol decoding and data splitting, and matches it with several small characteristics databases. It is shown in Fig. 2. First, detectors capture all network traffic, decode the data by protocol, obtain the protocol head field, and then use the detection engine to detect. We apply the normalized method in software engineering to explain the traditional IDS model. The normalized method in software engineering means categorization of events that are in need of treatment and which constitute an individual or a group of fixed and extended data structures and processing programs. The primary purpose of the IDS model is to adopt the normalized method in the case of network intrusion. As this is geared towards protocol analysis, the detection speed and degree of detail depend on three principal factors: the depth of protocol analysis, the speed of characteristic matching and the quality of the characteristic database, in which detection speed is inversely proportional to protocol analysis depth, and directly proportional to the speed of characteristic matching.

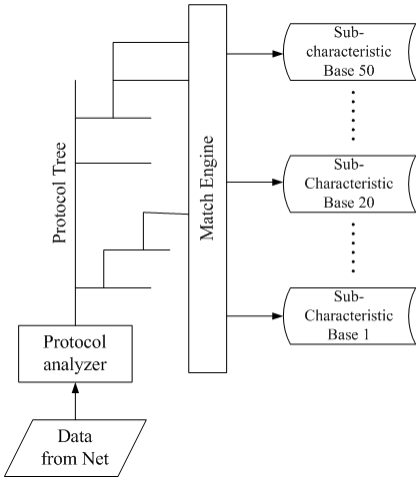

Figure 2. Traditional IDS architecture

The design guideline of classical network IDS is to minimize the rule set, after detailed protocol decoding, the rule set matching the least number is achieved. The matching speed has always been the bottleneck in the elementary phase of an IDS system. Since the relationship between the running time of the detection algorithm and that of the record number is linear, the most effective method for obtaining the minimum detection time is to decrease the number of characteristics to the greatest possible extent. With detailed protocol decoding, matching with a small scale characteristic database is possible and the detection speed can be increased. Currently, most network IDSs adopt a framework similar to Snort, namely, assigning thousands of rules to 40-60 small scale characteristic databases associated with protocol normalization.

The method is effective for generalized intrusion behavior on a low-speed network, but for some new security problems, such a framework has obvious shortcomings. The most serious problem is caused by network worms that affect the safety of global

networks. Network worms propagate through network media and cause serious damage to the network with its actions such as destroying data, disturbing the running of system, stealing information from an infected host, distributing attacks against other nodes and opening backdoors on computers [15]. As of late, new worm mechanisms are developed continuously and worms are generated endlessly. The amount of internet worms contained in HAMCLIB (HIT-Antiy Malware Lib) up to June 18, 2005 is enumerated in Table 2.

Table 2. Internet worm category and quantity statistic table

Style	Quantity
E-mail worm	2807
IM-worm	172
P2P-worm	1007
IRC-worm	715
Other	675
Total	5376

With the profusion of network attack behaviors, especially with the rampant spread of internet worms, traditional network IDSs increasingly cannot satisfy the requirements. This is because it is based on detailed protocol decoding using small-scale rule set matching.

From the viewpoint of network detection, worms can be classified according to two propagation paths, one is exploit-based, and the other is normal network transmission protocol based. The former can be detected using the shellcode it sends as characteristics, while the latter, such as mass-mailing worms and P2P worms, have no such general characteristics. For these worms, to detect them consistently is to find their transmission characteristic, i.e. the way to combine traditional virus characteristic extraction technology and network detection technology. However, the requirements as to the scale of the detection database are unprecedented. If network detection rules are to be fully formed, the capacity of the existing IDS framework will be outgrown.

3.2 High Speed Matching Algorithms and Applications in VDS

With the application of high-speed matching algorithms such as AC and BM in the IDS field, the problem of the linear relationship between the time needed for the detection algorithm and record number has been solved. To improve the performance of IDS, these matching algorithms can simply be incorporated into the current IDS framework. However, since the basic concept of detailed protocol decoding has become outgrown by the problem, the optimality of the entire framework has to be reanalyzed.

Two main problems exist in the traditional IDS model of protocol normalization: one is that time needed for decoding and developing the tree hierarchy using the traditional IDS model can hardly satisfy the requirements. The other is that network worms (including virus files transmitted via the network) may propagate through many different ways. Therefore, if transmission characteristics are given as rules, it will cause redundancy since rules with matching content will be needed for each different protocol.

Based on the idea put forward above, a new IDS normalized detection concept is adopted in this paper. The new IDS framework is designed on the basis of a high-speed matching algorithm. The new normalized model, with detailed, high-speed matching at its core is oriented toward algorithmic matching, not toward protocol analysis. In HAMCLIB, we apply VCC (Virus Characteristic Characterization, where virus is still used to denote Malware) to describe characteristics of Malware. VCC uses a set of descriptions of malware characteristics, which has a different description method for different virus characteristics.

An example is given to show the architecture of network worm detection; the VCC in the following section is the VCC description for backdoor tools Backdoor.bo in HAMCLIB.

alert
virus(type:”Backdoor”;name:”bo”;version:”a”;size:”124928” ;content=|81EC0805000083BC240C05000000535657557D148B8424240500008BAC242005000050E9950500000F85800500008B|)

Assume that Backdoor.bo is uploaded through FTP; then the VCC description can be converted to a Snort-style rule:

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Backdoor.bo.a Upload"; content: |81EC0805000083BC240C05000000535657557D148B8424240500008BAC242005000050E9950500000F85800500008B |;)

A detection rule protecting MS SQL Server in Snort’s rule set is taken as reference:

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433 (msg:"MS-SQL shellcode attempt"; content: "|48002500780077009000900090009000900003300c000500068002e00|"; flow:to_server,established; classtype:shellcode-detect; sid:693; rev:3;)

The transmission protocols of the two rules above are entirely different, but detection concepts are the same, both of their aims are to match a special characteristic string. Such rules can be treated by normalized method. Hence the data detected should not be classified according to protocol as in the traditional method, but should be constructed according to the framework of the matching algorithm.

Based on the algorithm detection model, data can be divided into three types: data needing deep content pre-treatment, data needing a special algorithm and data that can be directly matched at the binary level.

Data needing deep content pre-processing includes two kinds of data mainly: one is Web pages, which require detailed script worm detection. An online script compressor is required to dynamically detect script worms at network level. The other one is to pre-treat coded mail and recognize Email worms. Traditional IDS cannot perform detection in the two situations mentioned above. We need an algorithm to process case-insensitive data for cases such as a URL attack.

Therefore the new logic of the normalized model is to regard all network data, except for the two kinds mentioned above, as the kind that can be directly matched on binary level. Detailed protocol tree analysis is omitted in the new model, and only the most basic data flow split structure is preserved.

3.3 VDS: Modifying High Speed Matching Algorithms

Based on the description above, we have built a high-speed model structure, which has better performance than traditional IDS in detecting the transmission characteristics of internet worms and binary viruses. However the detection basis of the structure is content-matching events, which is not suitable for detecting data

without distinctive content characteristics or attack data with indistinctive content characteristics. Cases such as false alarms will occur easily. For the first case, the attack itself is judged directly by correlating the protocol norms with the characteristics of the data package. While for the second case, the protocol characteristics may be required as auxiliary information. The rule given as follows is to detect a DoS attack by the name of "DoS UDP echo+chargen bomb":

alert udp any 19 <> any 7 (msg:"DoS UDP echo+chargen bomb"; reference:cve,CAN-1999-0635; reference:cve,CVE-1999-0103; classtype:attempted-dos; sid:271; rev:3;)

We can find that content matching does not exist in the rule; characteristics of the attack are shown as port or protocol information. Since it has no distinctive content characteristics, it cannot be detected by the new structure described in the former section.

As can be seen from the analysis, the new normalized model can improve the performance of IDS, but it is unable to take on some of the other functionality of IDS. The simplest solution is to combine traditional IDS protocol analysis with high-speed matching algorithms so that matching speed is improved, with the result that:

Total time for disposal = Protocol decoding time + Matching time

Obviously, the total time for disposal is unacceptable. The model built in the above section has achieved independence of protocol from content matching on the basis of a most basic data split. We can modify the model by changing the serial relationship between protocol parsing and content matching to a parallel relationship. Then the protocol parsing is not preprocessing but detection. Therefore, the total time for disposal is:

Total time for disposal= max {Protocol decoding time，Matching time}

This model adopts improved protocol analysis technology, called protocol location technology. The ultimate purpose of traditional protocol analysis technology is to determine the detection branch; hence the data is expanded into a tree structure with multiple steps. While protocol location technology requires information associated with network communication – which acts as a judgment method or an aux judgment method – its analysis method is a single step. Finally, we will normalize the system rules as follows: Matching rules based on deep content pre-processing, matching rules based on a special algorithm, binary level matching rules, binary level rules requiring network information, and rules of pure network information.

The IDS model based on the above architecture is shown in Fig. 3, which has advantages over traditional model shown in Fig. 2 in the following ways: higher detection speed, greater rule capacity and extended ability to deal with several attack types that cannot detected by traditional IDS.

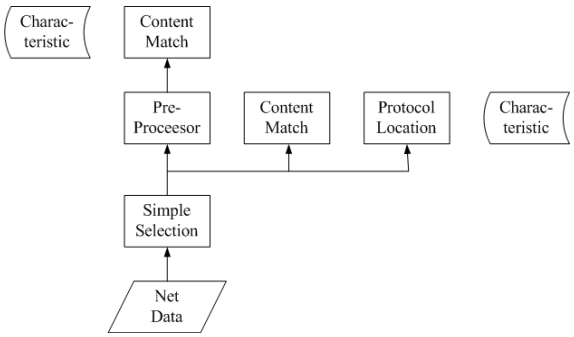

Figure 3. Improved IDS Architecture

4. SYSTEM REALIZATION

4.1 System Frame

According to the VDS model described in the above section, A Malware Detection System can be realized in this section. The design idea for system is: network data is mirrored to a special port of Ethernet switch; real-time network data flow is checked by the engine. Once a sensitive service is frequently accessed by worms or events accord with some characteristic, alarms are sent out immediately. At the same time, a detailed log is recorded for post audit and tracing. The VDS structure is shown in Fig.4

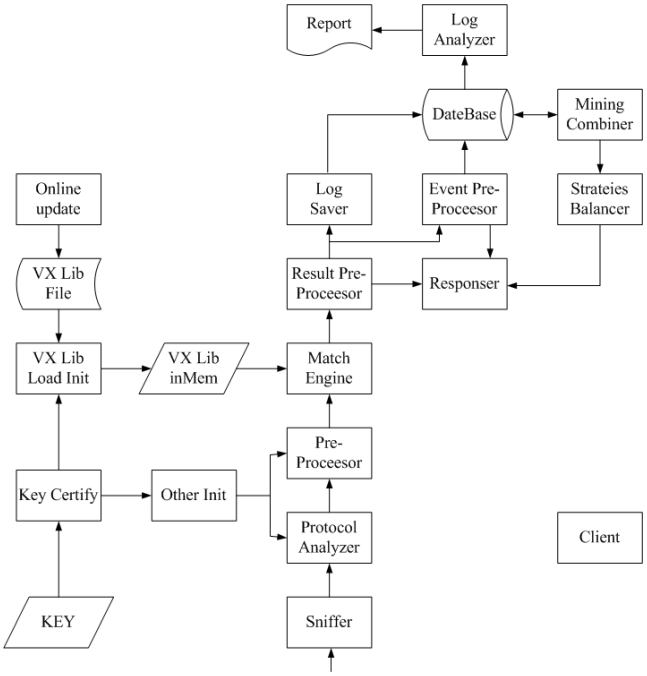

Figure 4. VDS Architecture

Not only can various malware and behavior taking advantages of network vulnerability be detected by such a system, but also the associated malware and detailed evidence of attack can be provided, which helps the strategy coordination center during the decision-making and security precautions setting process.

(In the above figure "Sniffer" is the module that captures traffic in the system. Zero-copy and parallel protocol stack [15] are adopted.)

4.2 Match Engine

The core modules of the Match Engine are four sub-detection engines: the Base Engine which is a static matching engine at the package level independent of protocol, the Mail Engine which is a static matching engine using high-speed pre-treatment technology, the Script Engine which is a script virus filtering engine using characteristics sequences and weighted combine, and the Url Engine which is a URL-scan detection engine working at the flow level.

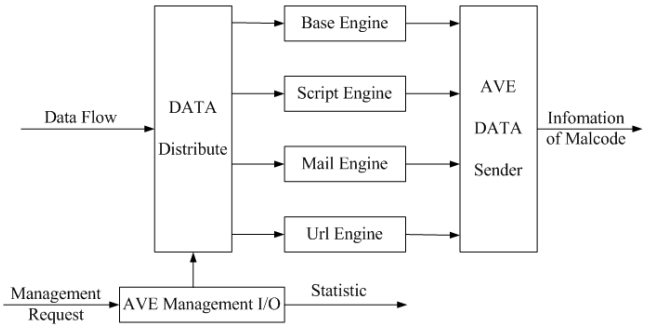

Figure 5. Match Engine Sketch Figure

The Match Engine adopts a normalized model for efficiency optimization, i.e. classifying network traffic data into four types: data matched at the binary level, case-insensitive data, data needing pre-treatment and data needing weighted matching, which are detected by the four detection engines respectively. The relation between match types and the corresponding detection engine are shown in Table 3.

Table 3. Correspondence between Matching Type and Engine

Match Model	Engine	Worm Style	Examples
Original Match	Basic Engine	Scan worm	Sasser
Insensitive Capital Match	URL Engine	WEB scan worm	CodeRed
Need Pre-processing Match	Mail Engine	I-worm	I-Worm, Nimda, Netsky
Weighted Match	Script Engine	Script worm	I-Worm, Happytime

Knowledge database files are formed accordingly, namely, character database of primary virus, Mail-worm, script virus and URL vulnerability. These characters can be divided into:

a. Transmission characteristics, which can be extracted from match rules based on the stream or packages when the worm body is transferred;

b. Scan characteristics, which can be extracted based on scanning the worm packets;

c. Attack (exploit) characteristics, which can be extracted based on shellcode used by the worm or other attack packages.

d. By means of cooperation with the packet sniffer module and detection module, original data is obtained as follows:

Mail-worm: name, transmission direction, corresponding IP quadruple, application protocol, accumulated transmission times per time unit, transmission traffic of single worm.

Scan worm: category or name, transmission direction, corresponding IP quadruple, application protocol, accumulated scan times per time unit, size of single scan package.

This data will be put in as the results of the pre-processor module and can be mined according to the flow diagram in Figure 4. Finally, the detection log can be created.

5. TEST RESULTS

Since May 2003, our test system has been working in CERNET of the Harbin Institute of Technology, the bandwidth of which is 1000Mbps, and in which 84 class C IP addresses are connected. With this system, worm body transmission (E-mail Worm), worm scanning, worm delivering, worm attacking and various other malware transmissions are detected successfully. As to the detection speed of the system: the binary engine is 2.5Gbps, the URL engine is 1.6Gbps, the mail engine is 0.8Gbps and the script engine is 0.3Gbps. There are 20,000 records in the system malware characteristics database. The system supplies online query functionality, which can query and keep record of malware passing the network in real time, and detection logs are displayed with per-minute data points. The system's real-time query window is shown in Fig.6 (the original interface is in Chinese, with English notation added later, same for the items that follow).

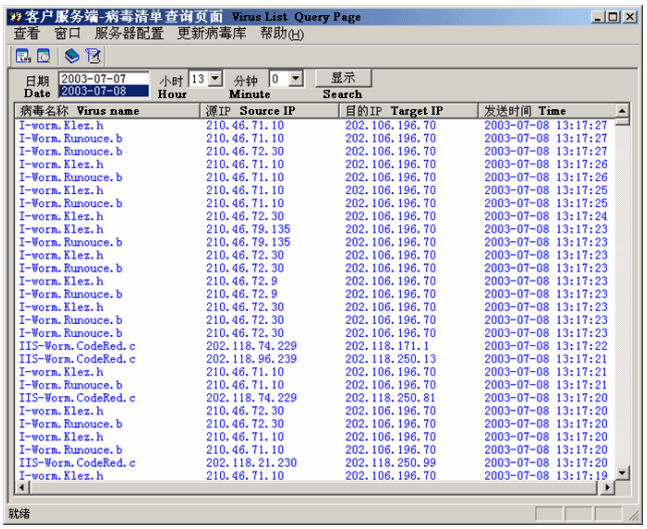

Figure 6. VDS system real-time query window

Users can query the log with an IP address or virus name. An IP address query of the log is shown in Fig.7, and a virus name query is shown in Fig.8.

The system provides various statistics functions in the Top 10 table and shows the Top 10 virus body transmission times, and the Top 10 virus scan times by day or month. The Top 10 table is shown in Figure 9.

The system can also track the total traffic flow generated by worm transmission through network, so as to calculate the amount of current bandwidth occupied by worms. Top 10 sources of Email-Virus Traffic are shown in Fig. 10.

From 202.118.0.0 To 202.118.255.255 Submit

Virus Name	Source IP	Target IP	Start Time	End Time
Worm.Win32.WelChina	202.118.253.16	118.118.235.53	2004-08-28	2004-08-28
RPC access	202.118.228.114	104.104.11.131	2004-08-28	2004-08-28
Worm.Win32.WelChina	202.118.253.16	118.118.237.18	2004-08-28	2004-08-28
RPC access	202.118.228.114	104.104.11.131	2004-08-28	2004-08-28
Worm.Win32.WelChina	202.118.253.16	46.46.79.57	2004-08-28	2004-08-28
RPC access	202.118.228.114	218.66.212.139	2004-08-28	2004-08-28
Worm.Win32.WelChina	202.118.253.16	217.217.250.5	2004-08-28	2004-08-28
RPC access	202.118.227.103	202.114.58.180	2004-08-28	2004-08-28
Worm.Win32.WelChina	202.118.253.16	217.217.250.59	2004-08-28	2004-08-28
RPC access	202.118.230.69	202.118.237.103	2004-08-28	2004-08-28

Figure 7. Search result according to IP range.

Virus Name netsky Submit

Virus Name	Source IP	Target IP	Start Time	End Time
I-Worm.NetSky.r	133.133.105.126	202.118.224.153	2004-08-04	2004-08-04
I-Worm.NetSky.r	118.118.234.201	202.118.224.153	2004-08-04	2004-08-04
I-Worm.NetSky.r	202.118.234.201	202.118.224.153	2004-08-04	2004-08-04
I-Worm.NetSky.r	18.18.45.74	202.118.224.153	2004-08-04	2004-08-04
I-Worm.NetSky.r	18.18.45.74	202.118.224.153	2004-08-04	2004-08-04
I-Worm.NetSky.aa	149.149.255.236	202.118.224.153	2004-08-04	2004-08-04

Figure 8. Search Result according to virus name

2003-7-7病毒扫描日志 Virus Log

统计数据: Total Number:		
扫描数据流总数:	0	(Amount of Scanning Data Flow)
发现病毒体总数:	242573	(Amount of Found Virus Body)
发现已知病毒总数:	242573	(Amount of Known Virus)
发现未知病毒总数:	0	(Amount of Unknown Virus)

发现病毒体传输次数排行榜: Top 10 of Virus Body Transmission Times

名次 No.	病毒名 Virus Name	发现次数 Found Times
1	I-worm.Klez.h	171267
2	I-Worm.Runouce.b	39661
3	I-Worm.Lentin.i	941
4	I-Worm.Lentin.m	167
5	I-Worm.Sobig.a	67
6	I-Worm.LovGate.f	54
7	I-Worm.Sobig.b	12
8	I-Worm.Sobig.c	2
9	I-Worm.Sobig	1
10	I-Worm.Tanatos.dam	1

发现病毒体扫描次数排行榜: Top 10 of Virus Scan Times

名次 No.	病毒名 Virus Name	发现次数 Found Times	源节点数 Num. of Source IP
1	IIS-Worm.CodeRed.c(Scan)	45513	1023
2	IIS-Worm(Scan)	980	2
3	I-Worm.Nimda(Scan)	577	109

Figure 9. Top 10 Table

Top 10 of Email-Virus Transmission Times

No.	Virus Name	Found Times	Traffic of Virus
1	I-Worm.NetSky.q	136	4021248
2	I-Worm.Runouce.b	37	4982124
3	I-Worm.LovGate.ad	33	4595712
4	I-Worm.NetSky.d	10	20510
5	I-Worm.NetSky.r	5	147840
6	I-Worm.LovGate.p	4	389120
7	I-Worm.Bagle.z	2	145266

Figure 10. Traffic of Virus

6. Discussion and Future Work

According to the processing procedure for malware epidemics based on the P2DR model given above, VDS can find the exact IPs of computers infected after obtaining the characteristics of the worm, which supplies accurate navigation information for the emergency response system. Thereby we can remotely remove the virus from the target computer by using management tools or remote removal tools. Since the system operates by monitoring bypassing traffic, we can disconnect the worm's connections with methods such as sending a reset message and so on. Cynthia Wong [6] et. al. consider that mail systems based on the SMTP protocol have the ability to re-send mails when the mail server has not received a response indicating that the messages were successfully sent, VDS will fabricate a response package to the sender when it disconnects a mail connection, so that the problem of flow increase due to connection interruption will not arise.

Since the amount of real-time match information obtained from a backbone network is quite significant, match information has to be dealt with, and information associated should be combined into events. Methods to combine events are given as follows:

a. Parallel-type combine method, which is to combine the records of different Source IPs, same Target IP, and same Type and ID into one event.. Typically, these are events sent by computers infected with email worms and events sent among mail servers.

b. Analysis-based Parallel combine, which means to combine and treat the events with same Source IP and Target IP, but different Type and ID – typically, a point-to-point scan event.

c. Radiant-type combine method, which combines the records with the same multiple Source IPs and different Target IPs. Generally this means the records of the Target IP can form a continuous IP range with the same Target Port, and same type and ID, typical of the behavior of worms or range-scanning by hackers.

d. Convergence-type combine method, which combines records with multiple different Source IPs and the same Target IP (including Target Port) – typical of DDoS attack behavior, worm online updates and connections to IRC. For its special characteristics, such detection functions should be combined with a white list to detect unknown worms.

e. Internal technology combine method, which implies treatment at inner systems, transparency to users, and processing repetitive match records caused by a segment of characteristics.

After combination of matching information, a great deal of data logging is cut down, the information supplied to emergency response systems is clearer, and serviceability is increased. Hereby in the future work will be focused on improving our research on combination and mining of matching results.

7. Conclusion

With more and more malware epidemics occurring, we propose a relatively ideal way to resolve this problem, which is a dynamic information security model: the P2DR model. The process should include the following stages:

1. The malware epidemic discovered by the detection system is sent to the strategy coordination center.

2. The strategy coordination center establishes a countermeasure strategy targeting the epidemic and transfers it to the protection and response system

3. The malware epidemic event is thwarted by protection and response in real-time, with the strategy adjusted dynamically to ensure the accordance of the protection and response system with the current network security state.

We have studied the detection process of the P2DR model in detail and have analyzed the structure of traditional IDS. We show that the traditional IDS architecture (which is based on detailed protocol decoding and split data, and matched with several small

characteristic databases) is unfit for monitoring malware epidemics on high speed network traffic and multivariant malware. Therefore, we have designed the match and protocol location analysis parallel model to normalize the detection rules of malware, and formed five kinds of rule sets, including:. match rules based on deep content pre-processing, match rules based on special algorithms, match rules for the binary level, match rules for the binary level requiring network information check and rules of pure network information. According to the parallel model and normalized rules database, the Virus Detection System is realized. Detection results of VDS are also discussed in this paper. We believe that it is advantageous to combine match results into events, and accordingly, presented five combination methods, including: parallel-type combine, parallel-type combine based on analysis, radiant-type combine, convergence-type combine and inter-technology combine.

8. REFERENCES

[1] Cheung S, Hoagland J, Levitt K, Rowe J, Staniford C, Yip R, Zerkle D. "The design of GrIDS: A graph-based intrusion detection system. Technical Report," *CSE-99-2, 1999.*

[2] John W. Lockwood, James Moscola1, Matthew Kulig, David Reddick, Tim Brooks, "Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware," *Military and Aerospace Programmable Logic Device, Sept. 2003.*

[3] Niemeyer, Robert E, "Applying the TNI to system certification and accreditation Reasoning about naming systems," *Annual Computer Security Applications Conference, 1989, p 248-252.*

[4] Zeng Zhifeng Yang Yixian, "On the Trend and Study of Network Security,". *Computer Engineering and Applications, (Chinese version) Oct, 2000.*

[5] David Moore, Colleen Shannon, Geoffrey Voelker, Stefan Savage, "Internet Quarantine: Requirements for Containing Self-Propagating Code," *Proceedings of the 2003 IEEE Infocom Conference, Apr. 2003.*

[6] Cynthia Wong, Chenxi Wang, Dawn Song, Stan Bielski, Gregory R. Ganger, "Dynamic Quarantine of Internet Worms." *The International Conference on Dependable Systems and Networks (DSN-2004),June, 2004.*

[7] Phillip Porras, Linda Briesemeister,Keith Skinner,Karl Levitt, Jeff Rowe, Yu-Cheng Allen Ting, "A Hybrid Quarantine efense," *WORM'04, Oct., 2004.*

[8] Frank Castaneda, Emre Can Sezery, Jun Xu, "WORM vs. WORM: Preliminary Study of an Active Counter-Attack Mechanism," *WORM'04, Oct., 2004.*

[9] Arno Wagner, Thomas Dubendorfer, "Experiences with Worm Propagation Simulations," *WORM'03, Oct., 2003.*

[10] Cliff Changchun Zou, Weibo Gong, and Don Towsley, "Code red worm propagation modeling and analysis," *In Proceedings of the 9th ACM Conference on Compute rand Communication Security, November 2002.*

[11] Yang Wang, Chenxi Wang, "Modeling the effects of timing parameters on virus propagation," *In Proceedings of the 2003 ACM workshop on Rapid Malcode, pages 61–66. ACM Press, 2003.*

[12] Cliff C. Zou, Don Towsley, Weibo Gong, "Email Worm Modeling and Defense," *13th International Conference On Computer Communications and networks, Oct. 2004*

[13] Cliff. C. Zou, Weibo Gong, and Don. Towsley, "Code red worm propagation modeling and analysis," *in Proceedings of 9th ACM Conference on Computer and Communications Security (CCS'02), Oct. 2002.*

[14] Zesheng Chen, Lixin Gao, Kevin Kwiat, "Modeling the Spread of Active Worms，" *In Proceedings of IEEE INFOCOM 2003.*

[15] Nicholas Weaver, Vern Paxon, Stuart Staniford, Robert Cunningham, "A Taxonomy of Computer Worms," *in Proceedings of the Workshop on Rapid Malcode, Oct., 2003.*

[16] Wang Bailing, Fang Binxing, Yun Xiaochun, "The Study and Implementation of Zero-Copy Packet Capture Platform," *Chinese Journal of Computers, Vol.28, No.1, Jan., 2005*.