Submit to: Workshop on Cryptology and Information Security

Paper Title: Detecting Distributed DoS/Scanning by Anomaly Distribution of Packet Fields

Chang-Han Jong, Shiuh-Pyng Shieh

{chjong,ssp}@csie.nctu.edu.tw

Contact author: Shiuypyng Shieh

Department of Computer Science and Information Engineering,

National Chiao-Tung University, Hsin-Chu, Taiwan

Phone: (03) 5744788

Fax: (03) 5734176

Abstract

To detect distributed denial of service and distributed scanning attacks, we propose Anomaly Dispersion Scheme (ADS). Observing the creation of packet fields in attack programs and normal programs, ADS monitors the distributions of packet fields, which alter when the attack programs using raw socket interface partake in, to detect intrusion. The sets of anomaly packet fields are attack signatures.

Keywords:

Distributed Denial of Service, Scanning, Network Intrusion Detection, Anomaly Detection

Detecting Distributed DoS/Scanning by Anomaly Distribution of Packet Fields

Chang-Han Jong, Shiuh-Pyng Shieh

{chjong,ssp}@csie.nctu.edu.tw

Department of Computer Science and Information Engineering,

National Chiao-Tung University, Hsin-Chu, Taiwan

Abstract

To detect distributed denial of service and distributed scanning attacks, we propose Anomaly Dispersion Scheme (ADS). Observing the creation of packet fields in attack programs and normal programs, ADS monitors the distributions of packet fields, which alter when the attack programs using raw socket interface partake in, to detect intrusion. The sets of anomaly packet fields are attack signatures.

Keywords:

Distributed Denial of Service, Scanning, Network Intrusion Detection, Anomaly Detection

1 Introduction

Distributed DoS and scanning (DDoS/DS) are among the most serious problems in network security. DDoS sends numerous malicious packets from multiple hosts to disable the victim hosts [Yu 90][Dittrich]. DS collects network information including live hosts, open ports, and vulnerable services by multiple hosts for future intrusion. The attackers may use various ways to conceal themselves. Table 1 lists the techniques: source IP spoofing hides the origin of the attack packets; destination IP spoofing conceals the true victims; TCP/IP protocol ambiguity, such as stealthy scanning or smurf, makes the attacks dissimulated; inter-protocol scanning uses one

protocol implementation, such as HTTP proxy, to scan other ports, such as FTP port; besides, attackers can perform attacks from multiple hosts so that tracing back becomes hard work.

Attack-Hiding Technique	Description
Source IP Spoofing	Source addresses do not belong to the attack hosts
Destination IP Spoofing	Destination addresses do not belong to the victim hosts by assigning small TTL.
TCP/IP Protocol Ambiguity	For packets not well-defined in specification, implementation may response in wrong way or doesn't respond
Inter-Protocol Scanning	Using one protocol implementation to scan the ports of other protocols.
Multiple Attack Host	Using thousands of hosts to attack

Table 1: Attack-Hiding Techniques

DDoS/DS remain traces on packet fields so that we can use for detection. For DDoS, fields of address and port field sprawl; for DS, packets with protocol vulnerability of design or implementation may be used to elude detection; Layer 3/4 attacks accompanied with DDoS/DS use special values of packet fields to perform attacks. Moreover, tracing the attack programs, we found that attack programs, which use raw socket interface, create network packets whose fields are one of the following three kinds-1) fixed value, 2) created by random function, and 3) created by a specified function. Table 2 shows sample code fragments of the three types. The three different kinds of packet fields are created by constant value 242, random() function of libc library, and k00kip() function of the attack program. The fields of attack packets have different distribution from normal traffic, because the attack packets are often created by raw socket interface instead of by the TCP/IP protocol stacks embedded in the operation system. Therefore, the distribution of network packet fields can be used to detect DDoS/DS attacks if DDoS/DS attack programs actually behave differently in

packet fields with other programs.

Generating of packet fields	Sample code fragment
Fixed value	*((u_short*)p_ptr)=htons(242); /*IP ID*/
Random function	ih.ip_id=htons(random()); /*IP ID*/
Certain function	ih.ip_sec.s_addr=k00kip(); /*IP Src Addr*/

Table 2: How Attack Programs Generate Packet Fields

Various related work have been done to detect DoS or scanning. GrIDS detects rapid malicious network activities by modeling the network activities [Staniford 95]. Packet aggregation watches the ICMP messages to detect failure of networks [Kanamaru 00]. SPICE detects scanning by the entropy concept and a correlation engine [Standiford 00]. IP addresses are not trustful if no authentication is applied. GrIDS and packet aggregation are fooled by the spoofed IP addresses. In addition, GrIDS computes the graph search, whose complexity grows exponentially with the size of networks. SPICE has now only been proved functional in detection scanning.

We proposed the Anomaly Dispersion Scheme (ADS), which detects DDoS/DS attacks on the assumption that Dos/DS attacks make distribution of packet fields different. ADS doesn't consider IP addresses as a trustful identifier of packet. Moreover, ADS assumes almost all packet fields can be forged, but nevertheless when forged fields distribution is far from the normal distribution, intrusion is detected. It's why ADS detects distribution DoS and scanning.

This paper is organized as follows. In Section 2, we present an Anomaly Dispersion Scheme for detecting DDoS/DS attacks. In Section 3, we show the experiments and discussion. And finally, Section 4 gives the conclusion.

2 Anomaly Dispersion Scheme

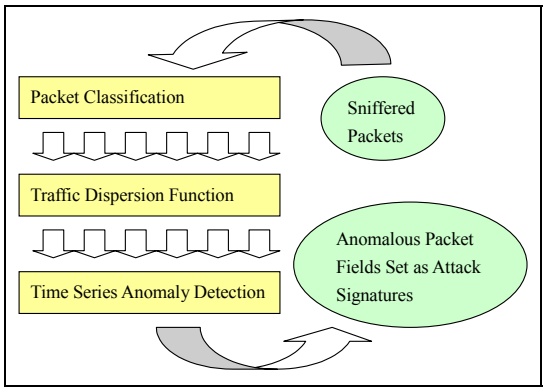

Figure 1: Overview of ADS

Figure 1 overviews ADS, which inputs sniffered packets and finally reports the anomalous packet fields as the attack signatures. ADS is composed of three components- 1) packet classification, 2) traffic dispersion function, and 3) time series anomaly detection. Packet classification is performed when ADS receives packets. Traffic dispersion function and time series anomaly detection are performed periodically.

Table 3 shows the pseudo code of ADS. Sniffered packets are passed to packet classification component to make the statistics, the packet count of each class (Line 8-11). For flexibility, class is defined as that two packets are in the same class if the values of the digests function of the two packets are the same. The packet digest function inputs one or more packet fields, or other properties of the packets and outputs a bit-reduced digest value.

Periodically, packets counts of classes at the current time period are then computed by the traffic dispersion function to present the composition the classes in a real number (Line 17). Subsequently, the time series, values of the traffic dispersion function in sequential time periods, are determined anomalous by the time series anomaly detection component (Line 18). If the time series is anomalous, it means that the

corresponding packet fields have anomalous distribution. It's notable that ADS concurrently uses multiple digest functions to monitor different packet fields (Line 9,15). In a period, ADS reports the set of the fields that has anomalous distribution as the attack signatures (Line 19).

There is environment dependent and configurable setting (Line 1-5). N packet digest functions are predefined to monitor different packet fields or packet properties. Packet digest functions should be defined according to the template between line 39-48. Detection window size decides how long should the current statistics should be compared the previous statistics. Parameter C4, the times of standard deviation of the historical profiles, specifies how much bias should the packet fields be considered anomalous.

```
 1 //***************  Environment Dependent and Configurable Setting  *************
 2 Predefined packet digest functions: PackDigest_i(), 0<i<n
 3
 4 Parameter C4, set it larger for higher false alarm rate and higher hit rate
 5 Parameter DWS, detection window size
 6
 7 //***************  Main  ************************************************
 8 When packets come, performing Packet Classification
 9       For packet digest function PacketDigest_c(), 0<c<n
10             bitstring digest=PacketDigest_c(packet p)
11             D_c,now_time,digest++
12
13 Periodically, performing Traffic Dispersion Function and Time Series Anomaly Detection
14       S={}
15       For packet digest function PacketDigest_c(), 0<c<n
16             //M_c,t refers to the set of D_c,t,i
17             TimeSeries TS={TDF(M_c,now_time-DWS), ...,TDF(M_c, now_time-1),TDF(M_c,now_time)}
18             If TSAD(TS)=ANOMOLIOUS then S=S ∪ {c}
19       Report S as attack signature
20
21 //***************  Subroutines  *****************************************
22 TDF(set X){        //Traffic dispersion function subroutine
23             Run only at 1st time period
24                   C1=1;C2=0;C3=1
25                   V= C3* ∑(C1*Log(x+1)+C2), where x belongs to X
26             Run only at 2nd time period
27                   C3= 1/V
28
29       G(X)=C3* ∑(C1*Log(x+1)+C2), where x belongs to X
30       return G(X)
31 }
32 TSAD(TimeSeries TS={s_1,s_2,s_3,...s_DWS}){        //Time Series Anomaly Detection subroutine
```

```
33        TimeSeries DTS={s2-s1, s3-s2,..., sDWS-sDWS-1 }
34        if mn> C4*StandardDeviation(DTS) then
35                  return ANOMOLIOUS
36        else
37                  return NORMAL
38 }
39 //***************   Template of Packet Digest Functions   ************************
40 PacketDigest(packet p){
41        bitstring digest
42        bitstring partial_digest
43        for each r, which belongs to the properties or packet fields of p we concern
44              partial_digest =bit_reduction(r)
45                    //bit_reduction() optionally uses group, aproximated, or hash mapping
46              digest =concat(digest, partial_digest)
47        return digest
48 }
```

Table 3: ADS Pseudo Code

2.1 Packet Classification

ADS receives packets from routers or via sniffering, and then classifies them in multiple digest functions. Two packets are regarded as the same class, if the results of packet digest function are the same. For packet digest functions, there are four kinds of input, including 1) packet fields, 2) length of the header and packet, 3) integrity and validity of the header or packet, 4) true properties (true packet length, not the recorded one).

If the maximum possible class number is too large, ADS may fail because of exhausting search. To reduce the maximum possible class number, bit reduction should be performed. There are three kinds of bit reduction techniques can be optionally adapted, 1) group mapping, 2) approximated mapping, and 3) hash mapping. Group mapping reduces the output bit according the semantics of the packet fields. For example, the network ID represents the full 32bits IPv4 address and ports are separated into well-known and large-than-1024 ones. Approximated mapping, such as dividing or modulation, is used when the packet fields represent the length. TCP SYN/ACK sequence numbers are also feasible to be used in approximated

because a TCP session is supposed to use sequential numbers. Hash grouping, such as MD5 algorithm or simple modulation is used.

2.2 Traffic Dispersion Function

Traffic dispersion function inputs packets counts of classes, and outputs a real value to present the composition. For example, there are 13 sniffered packets classified to 4 classes, whose packet counts are {3,4,2,4}. Then the traffic dispersion function inputs {3,4,2,4} and outputs a real value to represent {3,4,2,4}. Traffic dispersion function is supposed to distinguish network traffic with and without DDoS/DS attacks. That is, values of traffic dispersion function with and without DDoS/DS attacks should differ as much as possible

To make the ADS detect DDoS/DS well, the traffic dispersion function is supposed to be 1) aggregative, 2) insensitive, and 3) balanced. Without loss of generality, for packets counts $\{s_1,s_2,..,s_k\}$, aggregative is that traffic dispersion function should reflect the whole change but not fractions change, that is, if only a small subset of packets counts change few, then the traffic dispersion function value should almost be the same. Insensitive is that traffic dispersion function correlates to the quantities of packet counts (s_i, $1 \leqq i \leqq k$), not only the ratio ($s_i / \sum s_j$, $1 \leqq i,j \leqq k$). Insensitive resists from that the attackers to make the injected attack packet with the same distribution of packet fields with the normal traffic so that attacks are not detected. Balanced is that extreme quantity of certain class count would not affect value of traffic dispersion function too much. Balanced is to reduce the extreme big or small quantities of few class counts cause the false alarm rate high. There are extreme examples of traffic dispersion function that are not balanced. An IDS uses $f(X)=\sum(1/Pr(x))$, where x belongs to X, as the malicious function, where Pr(x) is the probability of the event. In

this case, if Pr(x) is small, then f(x) is significant large. When normal events are comparably with low probability, it may cause the IDS to have high false alarm rate. Another case is the second-order movement, $f(X)=\sum x^2$, where x belongs to X, which have extreme large value when the packet count of certain classes are high.

If a function g() is positive, increasing and its order is between (0,1), then $G(X)=\sum g(x)$, where x belongs to X, is a traffic dispersion function that satisfies aggregative, insensitive, and balanced. We propose a traffic dispersion function based on the above description: $G(X)=C3*\sum(C1*Log(x+1)+C2)$. The order of Log(x) is between (0,1), increasing and positive when x>1. So x+1 adjusted the function to satisfy the require properties. C1 controls the order and can be modified to adapt different environments. C2 gives weights those classes that have zero packet count. C3 normalizes the output of G(X), and is for privacy issue when opening the use of statistics in to public (Line 23-27 on Table 3). C1 and C2 are now fixed to 1 and 0 (Line 24 on Table 3). Dynamic decided C1 and C2 may be provided in the future.

2.3 Time Series Anomaly Detection

The differencing technique is used to reduce seasonal effect and trend in time series analysis [Chatfield 89]. First-order differencing transforms a time series $\{x_1,...,x_n\}$ to another time series $\{y_1, ...,y_{n-1}\}$ by performing $y_t=x_{t+1}-x_t$. This technique has also been used in detection of VoIP traffic anomaly [Mandjes 00]. Therefore this component uses this technique to process the time series produced from the traffic dispersion function component. Denning's Mean and Standard Deviation Model variation with differencing is used in this component for simplicity [Denning 86]. The physical meaning is that, rather than forecasting, the concept of variance tolerance is used to decide anomaly because we may not know how packet fields are used.

3 Experiments and Discussion

We have implemented a prototype based on libpcap packet capturing library and FreeBSD 4.4(x86) operation system. Samples are tcpdump packet captures from live network traffic, without or with injected attack. The traffic is captured on the gateway of a campus class C LAN with over 200 computers inside. Injected attacks are real-time performed by three famous DDoS/DS tools, stacheldraht, nmap and ping. Each sample lasts 60 seconds and totally 145 samples (98 without injected attacks) is tested. The injected attacks start at the 30th second. When processing, ADS skips the first 6 seconds because captured packets bursts in the first few seconds due to libpcap initialization. The detection windows size is 8 and will be discuss on the end of this section.

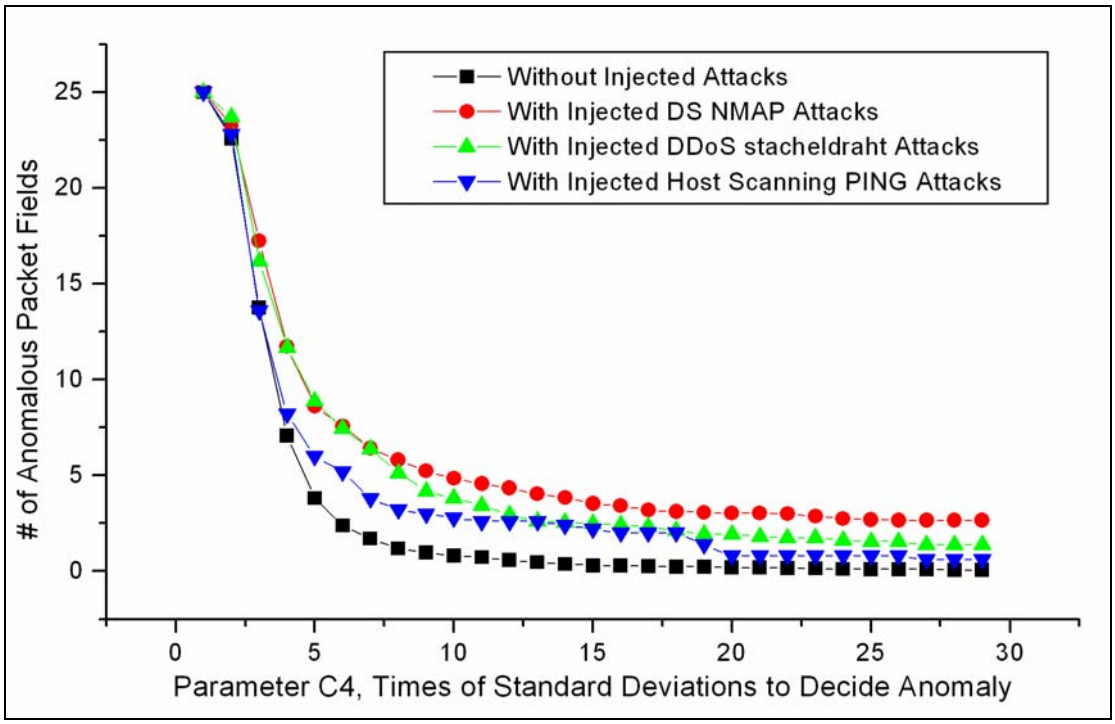

Figure 2: Average Number of Anomalous Packet Field

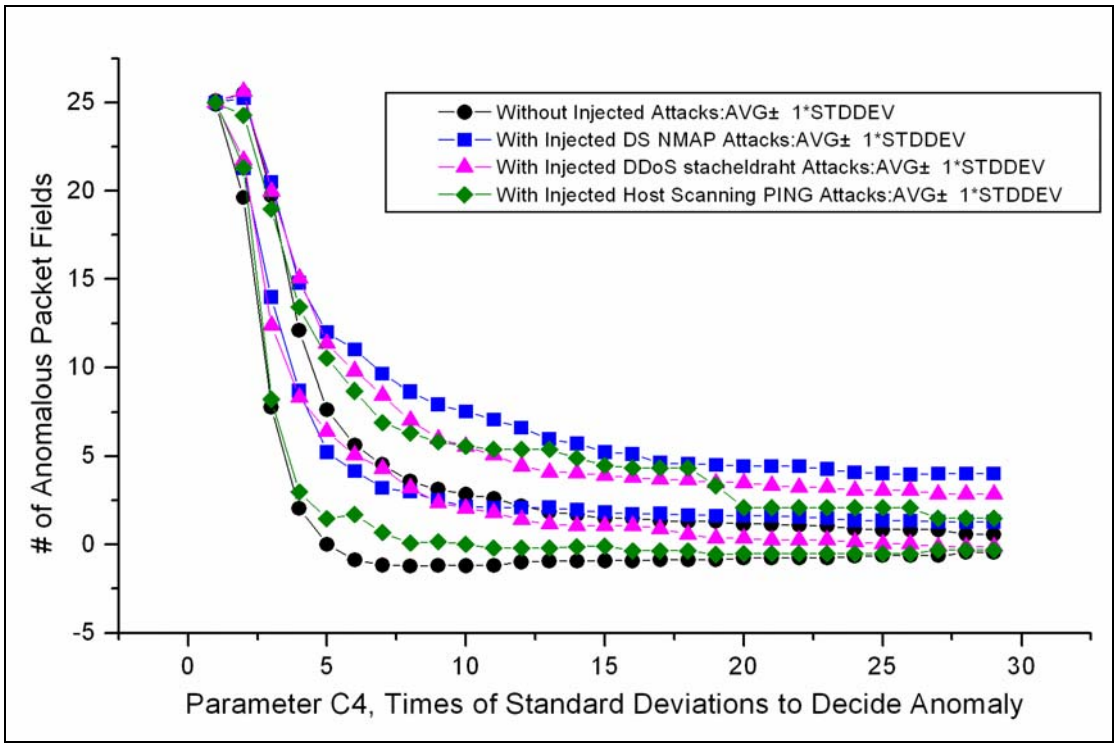

Figure 3: Number of Anomalous Packet Field: AVG±STDDEV

Figure 2 shows the average number of anomalistic packet fields with and without injected DDoS/DS attacks. X-axis is the tunable C4 threshold, which specifies how statistics bias is treaded as anomalistic. Lesser the C4, higher the hit rate and false alarm rate. In the Figure 2 we can see that under the same threshold C4, the average number of anomalous packet fields without injected DDoS/DS attacks is lower than the number with injected DDoS/DS attacks.

Figure 2 is not sound because only average number anomalistic packet fields is shown; Figure 3 completes it by drawing average number plus and minus 1 standard deviation of the anomalous packet fields. Two lines with the same symbol (circle, square, upper triangle, or diamond) specify the band of (AVG+STDDEV,AVG-STDDEV) to the number of anomalous packet fields. We can see that the band of "*With Injected DS NMAP Attack"*" and "*With Injected DDoS stacheldraht Attacks*" almost don't overlap "*Without Injected Attacks*". It means that DDoS stacheldraht Attacks and DS NMAP

Attacks can be detected effectively. The band of "*With Injected Host Scanning PING Attacks*" overlaps the band of "*Without Injected Attacks*". It refers that PING is not well detected because PING is used frequently in normal traffic without injected attacks.

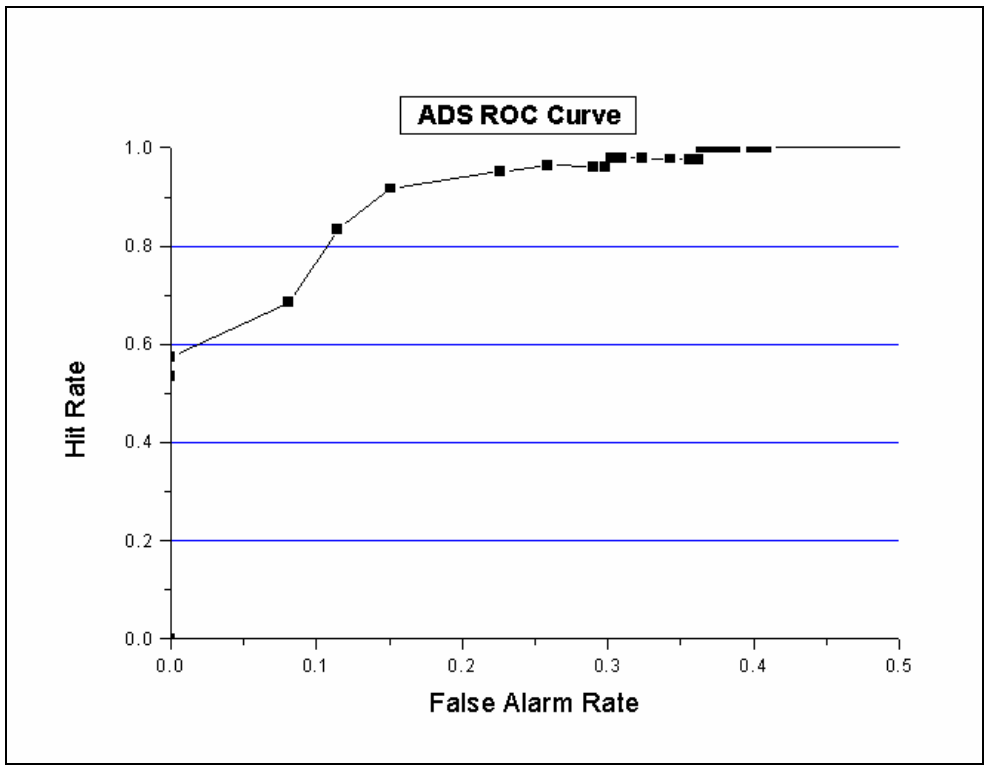

Figure 4: ROC Curve of ADS

The throughput of the ADS prototype, running on Pentinum III processor, with 25 packet digest functions is 102K Packets/second. Figure 4 shows the R.O.C curve of the ADS. ROC is an IDS performance evaluation graph, where the X-axis is the false alarm rate (false alarms/negative samples) and Y-axis is the hit rate (correct alarm sample/total alarm samples). The false alarm rate is not low because no probability-based or data-mining techniques are used to reduce the false alarm rate in our scheme.

This testing flow can be used to discover novel attacks. During the experiments, one type of novel network attack based on Microsoft Windows Universal PnP protocol

was discovered.

3.1.1 Anomaly Distribution of TCP/IP Packet Fields

ADS alarms for network attacks as attack signatures. We conclude the reasons why certain fields are anomalistic, which helps security officers to identify why the packet fields anomaly. Table 6 shows the reasons of anomalous packet field distribution. IP protocol fields set to 255, the reserved protocol, may be used to elude detection. IP ID generated by various algorithms, such as sequential counter, MD4, and MD5, would have different distribution than fixed valued. IP Offset field is anomalistic when small fragment packets for dissimulating or MTU is not suitable. IP and TCP Options field is used for information discovery for a long time. TCP SEQ/ACK # field is supposed to be sequential or near sequential (TCP may be out of order, but still controlled by sliding window). If the attack program uses fixed value on these fields, then anomaly is predictable.

Besides the single packet field anomalous distribution, OS fingerprinting attacks may almost all packet field distributions anomalous. OS fingerprinting is performed by sending a series of TCP/IP packets and receiving the responses of the target hosts. Therefore rare field values are presented in the OS fingerprinting packets. For attackers to increase the accuracy of OS fingerprinting, they don't use only a few packet fields, OS fingerprinting reveals anomaly in almost all fields.

Packet Field	The reason of anomalous distribution
IP Protocol	Stacheldraht tool sets this field to fixed value 255, which is the reserved value so that the author of the program may want to elude somehow. IP packets, whose protocol not equal to TCP/UDP, may be combined used with IP options to perform scanning.
IP ID	Older Un*x system use a sequential counter to generate IP ID fields; Linux uses MD4 algorithm; FreeBSD uses old fashion sequential counter or MD5 algorithm. So the IP ID field easily presents anomaly in attacking if the distribution of IP ID fields differ

IP Offset	When a packet is fragmented, IP offset field records the offset of spited packets related to the original packet. If the attack program wants to use tiny fragment packet to hide its activity, the IP offset field will be anomalistic. Fragmentation also happens when the MTU of a network in the routing path is small then the MTU of other network in the routing path.
IP Options	IP Options can be use to get information about routing and time or to use source routing to detour the firewall.
ICMP	Attack programs may use ICMP to detect if the target host alive. ICMP are used to test if the target host alive before scanning. Therefore, ICMP packets reveal another kind of attack signature other than the main attack packets. But ICMP happens too frequently, it makes high false alarm rate.
TCP/UDP Port	By Stantiford's observation, Port scanning is to gather enough information from the port. Therefore scanning attacks may probably cause TCP or UDP port field distributions change.
TCP SEQ/ACK #	If the attack programs send special TCP packets to scan or DoS, the TCP SEQ/ACK # fields change enormously because TCP connection semantics are different from the normal ones. When the TCP connections are anomalistic, TCP SEQ/ACK number may have abnormal distribution. The normal semantics of TCP SEQ/ACK # is sequential, and if the attack programs uses fixed value, anomaly is detected.
TCP Options	Window scale factor may be use to on degrading the performance of TCP sliding window. TCP options can also be used for gathering information just as IP options.

Table 6: Reasons of Packet Fields Anomaly Distribution

3.1.2 Detection Window Size versus Seasonal Effects

Detection window size is the size of time series in the TSA. The detection window size affects the accuracy of the IDS. We observed that if periodically and the detection window size is smaller than the cycle, false alarm rate will be high because the periodical network activities may be impulse. For example, Microsoft windows operation system broadcasts about every ten seconds.

A time series {Xn}, has a periodical action in cycle L. Without the loss of generality, the time series is denoted $\{m_1, m_2, m_3,...m_L, m_{L+1}, m_{L+2}, ...,m_{2L},...\}$. Broadcast packets are on m_{iL},i is positive integer. When the detection windows <L, the samples may be between m_{iL} and $m_{(i+1)L}$ and doesn't contain any m_{iL}. If m_{iL} is much larger than m_{jL+k}, where 0<k<L, forecasting according the detection windows lapses. If the detection

windows size is large or equal than L, one of the m_{iL} will be included in the forecasting, false alarm rate reduces.

4 Conclusion

ADS derives from tracing and comparing the TCP/IP protocol stacks and DDoS/DS attack programs. The idea comes from that packets created by different programs differ in packet fields especially between native TCP/IP embedded the OS and DDoS/DS attack programs using raw socket interface. The R.O.C curves and the throughput of the prototype show that ADS is practicable. ADS can be easily extend its ability for monitoring more fields by only adding packet digest functions for the packet fields that we are interested in.

Because ADS uses primitive Denning's Mean and Standard Deviation Model variation to determine the anomaly of time series, and no correlation engine or probability decision mechanism in it, false alarm rate is not low. We would like to replace add these features to ADS in the feature.

Reference

[Chatfield 89] C. Chatfield, "The Analysis of Time Series-An Introduction 3rd Edition, " page 21, 1989

[Denning 86] Denning, "An Intrusion Detection Model," IEEE Trans. on Software Engineering 1986

[Dittrich] Dave Dittrich, Distributed Denial of Service (DDoS) Attacks/tools homepage, http://staff.washington.edu/dittrich/misc/ddos/

[Kanamaru 00] Kanamaru, "A Simple Packet Aggregation Technique for Fault Detection," Int. Journal of Network, 2000

[Mandjes 00] M. Mandjes, I. Saniee, and A. Stolyar, "Load characterization, overload prediction, and load anomaly detection for voice over IP traffic," Proceedings 38th Allerton Conference, Urbana-Champaign, US, pp. 567-576.

[Staniford 95] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle, “GrIDS-A Graph Based Intrusion Detection System for Large Networks,” National Information Systems Security Conference, 1996

[Staniford 00], Stuart Staniford, James A. Hoagland, Joseph M. McAlerney, ”Practical Automated Detection of Stealthy Portscans, ” ACM Workshop on IDS, 2000

[Yu 90] Che-Fn Yu, Virgil D. Gligor, “A Specification and Verification Method for Preventing Denial of Service,” IEEE Trans. on Software Engineering, Vol 16, No 6, June 1990