

入侵检测系统部署实验报告

班级： 学号： 姓名：

一、实验名称

二、实验目的

理解入侵检测系统的结构和工作原理

掌握入侵检测系统 snort 部署方法

掌握基本的 snort 规则编写方法

三、实验内容与要求

构建一个基于 Windows 平台的入侵检测系统

入侵检测系统 snort

后台数据库

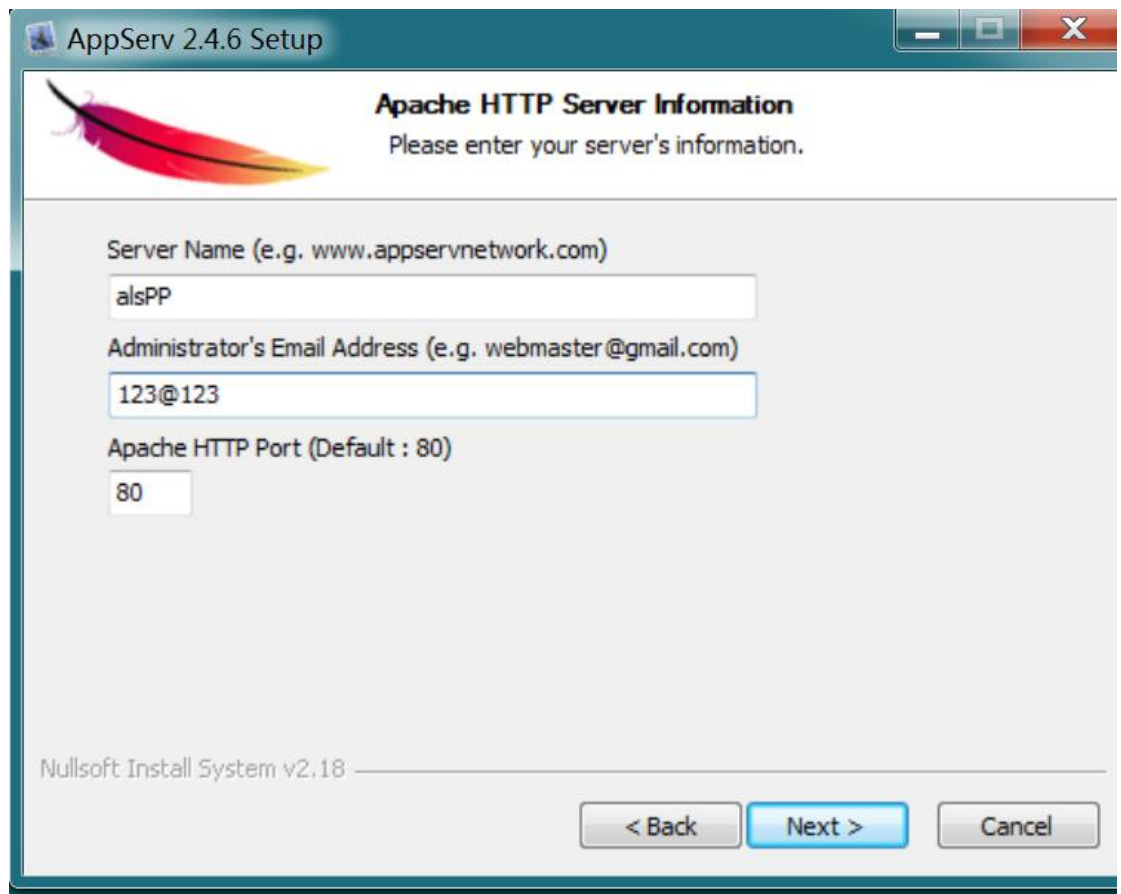
前端图形化分析显示

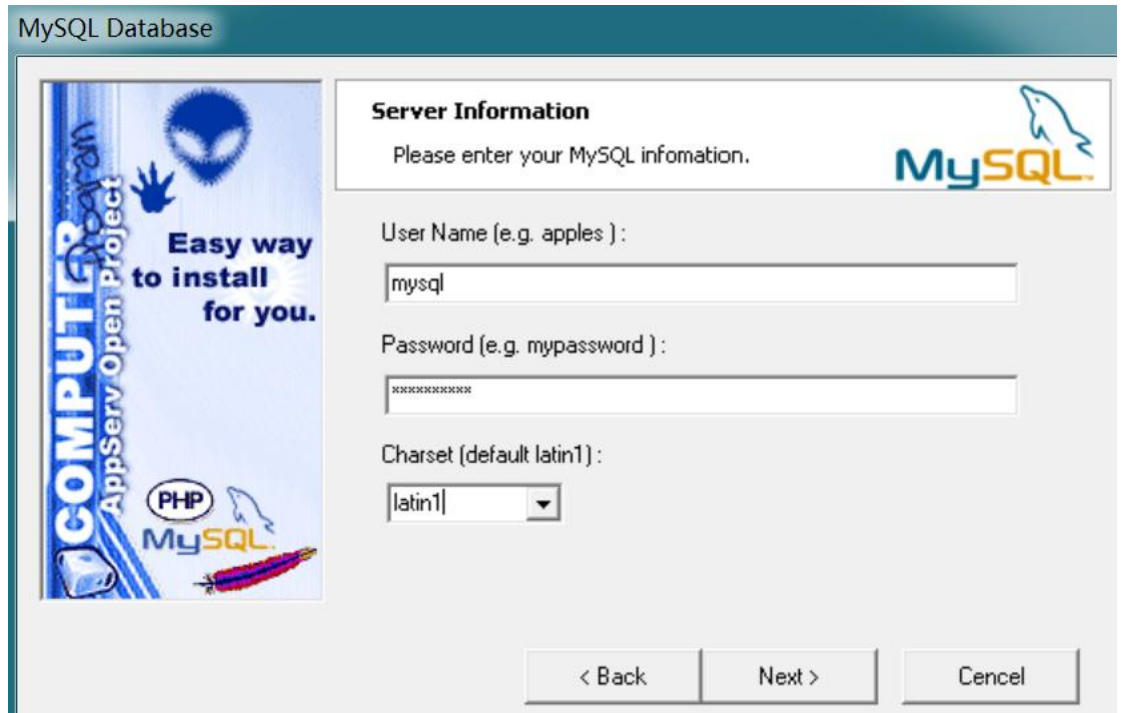
设计几条 snort 入侵检测规则，测试检测结果

四、实验过程

实验的详细步骤，过程

- WinPcap_3_0.exe(数据包捕获)
- Snort_2_3_0_Installer.exe (入侵检测系统)
- appserv-win32-2.4.1.exe (快速建立 Apache/PHP/MySQL)





- acid-0.9.6b23.tar.gz (入侵检测数据库分析控制台)
- adodb461.zip (PHP 数据库链接库)
- jppgraph-1.17.tar.gz (图形链接库 For PHP)



AppServ 安装成功

192.168.121.131 >> localhost >> snort | phpMyAdmin 2.6.0-rc1 - Windows Internet Explorer

http://192.168.121.131/phpMyAdmin/index.php

收藏夹 | 建议网站 | 网页快讯库

192.168.121.131 >> localho...

服务器: localhost 数据库: snort

您运行的 SQL 语句已经成功运行了。

SQL 查询:

```
# Copyright (C) 2000-2002 Carnegie Mellon University
#
# Maintainer: Roman Danyliw <rdd@cert.org>, <roman@danyliw.com>
#
# Original Author(s): Jed Pickel <jed@pickel.net> (2000-2001)
# Roman Danyliw <rdd@cert.org>
# Todd Schrubbs <ts@cert.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
CREATE TABLE schema(
  vseq INT UNSIGNED NOT NULL ,
  ctime DATETIME NOT NULL ,
  PRIMARY KEY ( vseq )
); # MySQL 返回的查询结果为空(即零行)。
INSERT INTO schema( vseq, ctime )
VALUES (
```

数据库: snort (16)

snort

- data
- detail
- encoding
- event
- icmphdr
- iphdr
- opt
- reference
- reference_system
- schema
- sensor
- sig_class
- sig_reference
- signature
- tcphdr
- udphdr

Snort 数据库脚本执行成功



数据库:

snort (16)

snort

- data
- detail
- encoding
- event
- icmp_hdr
- ip_hdr
- opt
- reference
- reference_system
- schema
- sensor
- sig_class
- sig_reference
- signature
- tcp_hdr
- udp_hdr

服务器: localhost

-
-
-
-
-
-

添加新用户

登入信息

用户名:

主机:

密码:

重新输入:

全局权限

注意: MySQL 权限名称会以英文显示
[全选](#) [全部不选](#)

数据	结构	管理
<input type="checkbox"/> SELECT	<input type="checkbox"/> CREATE	<input type="checkbox"/> GRANT
<input type="checkbox"/> INSERT	<input type="checkbox"/> ALTER	<input type="checkbox"/> SUPER
<input type="checkbox"/> UPDATE	<input type="checkbox"/> INDEX	<input type="checkbox"/> PROCESS
<input type="checkbox"/> DELETE	<input type="checkbox"/> DROP	<input type="checkbox"/> RELOAD
<input type="checkbox"/> FILE	<input type="checkbox"/> CREATE TEMPORARY TABLES	<input type="checkbox"/> SHUTDOWN
		<input type="checkbox"/> SHOW DATABASES
		<input type="checkbox"/> LOCK TABLES
		<input type="checkbox"/> REFERENCES
		<input type="checkbox"/> EXECUTE
		<input type="checkbox"/> REPLICATION CLIENT
		<input type="checkbox"/> REPLICATION SLAVE

资源限制

注意: 将这些选项设为 0 (零) 将删除限制。

MAX QUERIES PER HOUR

MAX UPDATES PER HOUR

MAX CONNECTIONS PER HOUR

执行

用户 'SnortUser'@'%' - 数据库 snort

• 编辑权限

按数据库指定权限		
注意: MySQL 权限名称会以英文显示		
全选 全部不选		
数据	结构	管理
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP	
	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	

• 按表指定权限

表	权限	授权	按列指定权限	操作
			无	

在下列表添加权限:

分配权限成功

• 按数据库指定权限

数据库	权限	授权	按表指定权限	操作
snort	ALL PRIVILEGES	是	否	编辑 收回

在下列数据库添加权限:

配置 acid, 修改完后打开 http://localhost/acid/acid_db_setup.php 报错

```
set allow_call_time_pass_reference to true in your INI file. However, future versions may not support this any longer. in
c:\appserv\www\acid\acid_state_citems.inc on line 1181
```

Warning: Call-time pass-by-reference has been deprecated - argument passed by value; If you would like to pass it by reference, modify the declaration of [runtime function name](). If you would like to enable call-time pass-by-reference, you can set allow_call_time_pass_reference to true in your INI file. However, future versions may not support this any longer. in c:\appserv\www\acid\acid_state_citems.inc on line 1216

Warning: Call-time pass-by-reference has been deprecated - argument passed by value; If you would like to pass it by reference, modify the declaration of [runtime function name](). If you would like to enable call-time pass-by-reference, you can set allow_call_time_pass_reference to true in your INI file. However, future versions may not support this any longer. in c:\appserv\www\acid\acid_state_citems.inc on line 1216

Warning: Call-time pass-by-reference has been deprecated - argument passed by value; If you would like to pass it by reference, modify the declaration of [runtime function name](). If you would like to enable call-time pass-by-reference, you can set allow_call_time_pass_reference to true in your INI file. However, future versions may not support this any longer. in c:\appserv\www\acid\acid_state_citems.inc on line 1250

Warning: Call-time pass-by-reference has been deprecated - argument passed by value; If you would like to pass it by reference, modify the declaration of [runtime function name](). If you would like to enable call-time pass-by-reference, you can set allow_call_time_pass_reference to true in your INI file. However, future versions may not support this any longer. in c:\appserv\www\acid\acid_state_citems.inc on line 1351

Warning: Call-time pass-by-reference has been deprecated - argument passed by value; If you would like to pass it by reference, modify the declaration of [runtime function name](). If you would like to enable call-time pass-by-reference, you can set allow_call_time_pass_reference to true in your INI file. However, future versions may not support this any longer. in c:\appserv\www\acid\acid_state_citems.inc on line 1351

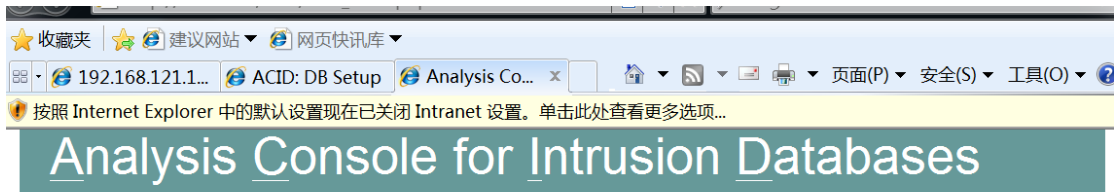
ACID DB Setup [Home](#) [Search](#) | [AG Maintenance](#)

[Back]

Error loading the DB Abstraction library: from "C:\AppServ\www\acid\adodb.inc.php"

Check the DB abstraction library variable \$DBlib_path in acid_conf.php

The underlying database library currently used is ADOdb, that can be downloaded at <http://php.weblogs.com/adodb>



Added 0 alert(s) to the Alert cache

Queried on : Tue May 06, 2025 22:51:47
Database: snort@localhost (schema version: 106)
Time window: no alerts detected

Sensors: 0 Unique Alerts: 0 (0 categories) Total Number of Alerts: 0 <ul style="list-style-type: none">Source IP addresses: 0Dest. IP addresses: 0Unique IP links 0Source Ports: 0<ul style="list-style-type: none">TCP (0) UDP (0)Dest. Ports: 0<ul style="list-style-type: none">TCP (0) UDP (0)	Traffic Profile by Protocol TCP (0%) UDP (0%) ICMP (0%) Portscan Traffic (0%)
--	--

- [Search](#)
- [Graph Alert data](#)

• **Snapshot**

- Most recent Alerts: [any protocol, TCP, UDP, ICMP](#)
- Today's: alerts [unique, listing](#); IP [src / dst](#)
- Last 24 Hours: alerts [unique, listing](#); IP [src / dst](#)
- Last 72 Hours: alerts [unique, listing](#); IP [src / dst](#)
- Most [recent 15 Unique Alerts](#)
- Last Source Ports: [any, TCP, UDP](#)
- Last Destination Ports: [any, TCP,](#)
- Most [frequent 5 Alerts](#)
- Most Frequent Source Ports: [any, TCP, UDP](#)
- Most Frequent Destination Ports: [any, TCP, UDP](#)
- Most frequent 15 addresses: [source, destination](#)

```
命令提示符 - snort -v
05/07-15:01:40.466154 192.168.120.205 -> 224.0.0.22
PROT0002 TTL:1 TOS:0x0 ID:4437 IpLen:24 DgmLen:40
IP Options <1> => Opt 148: 00 00
+++++
05/07-15:01:40.466676 192.168.120.205:54873 -> 224.0.0.252:5355
UDP TTL:1 TOS:0x0 ID:19597 IpLen:20 DgmLen:61
Len: 33
+++++
05/07-15:01:40.542859 192.168.120.205 -> 224.0.0.22
PROT0002 TTL:1 TOS:0x0 ID:4438 IpLen:24 DgmLen:40
IP Options <1> => Opt 148: 00 00
+++++
05/07-15:01:40.886603 192.168.120.205:54873 -> 224.0.0.252:5355
UDP TTL:1 TOS:0x0 ID:19598 IpLen:20 DgmLen:61
Len: 33
+++++
05/07-15:01:42.455704 192.168.120.205 -> 224.0.0.22
PROT0002 TTL:1 TOS:0x0 ID:4439 IpLen:24 DgmLen:40
IP Options <1> => Opt 148: 00 00
+++++
```

停止后拿到统计信息

```
Snort received 41 packets
Analyzed: 41(100.000%)
Dropped: 0(0.000%)

=====
Breakdown by protocol:
TCP: 1 (2.439%)
UDP: 10 (24.390%)
ICMP: 0 (0.000%)
ARP: 0 (0.000%)
EAPOL: 0 (0.000%)
IPv6: 20 (48.780%)
IPX: 0 (0.000%)
OTHER: 10 (24.390%)
DISCARD: 0 (0.000%)

=====
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
```

五、实验结果分析

基于网络的入侵检测系统（NIDS）：监控网络流量，寻找已知的攻击模式或异常行为。

- 信息收集系统：收集系统、网络、数据及用户活动的状态和行为。
- 信息分析系统：分析收集到的信息，通过模式匹配、统计分析和完整性分析等技术手段检测入侵。
- 响应：根据检测结果采取相应的响应措施。

The screenshot shows the Snort Analysis Console for Intrusion Databases interface on the left and a Windows PowerShell terminal on the right. The console displays various statistics and traffic profiles. The terminal shows the results of a ping command to 192.168.120.200, indicating successful connectivity with 4 packets received and 0 lost.

规则:

alert udp any any -> 192.168.164.131/24 1:1024 (msg:"UDP flow to low port");

测试:

#	Time	Source IP	Destination IP	Protocol
#130 (1.95)	2025-05-22 16:50:02	192.168.164.131	192.168.164.131	UDP
#131 (1.85)	2025-05-22 16:50:24	192.168.164.131	192.168.164.131	UDP
#132 (1.96)	2025-05-22 16:50:09	192.168.164.131	192.168.164.131	UDP
#133 (1.96)	2025-05-22 16:53:25	192.168.164.131	192.168.164.131	UDP
#134 (1.96)	2025-05-22 16:53:28	192.168.164.131	192.168.164.131	UDP
#135 (1.77)	2025-05-22 16:53:23	192.168.164.131	192.168.164.131	UDP

nc -u 192.168.164.131 123

alert tcp any any -> 192.168.164.131/24 1:1024 (msg:"TCP flow to low port");

测试:

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#545 (1.434)	TCP flow to low port	2025-05-22 17:20:30	192.168.164.132:5000	192.168.164.131:23	TCP
#546 (1.435)	TCP flow to low port	2025-05-22 17:20:30	192.168.164.132:5000	192.168.164.131:23	TCP
#547 (1.436)	TCP flow to low port	2025-05-22 17:20:30	192.168.164.132:5000	192.168.164.131:23	TCP
#548 (1.437)	TCP flow to low port	2025-05-22 17:20:19	192.168.164.132:5000	192.168.164.131:23	TCP
#549 (1.488)	TCP flow to low port	2025-05-22 17:20:19	192.168.164.132:5000	192.168.164.131:23	TCP

nc 192.168.164.131 80

alert tcp 192.168.164.131/24 any <> 192.168.164.131/24 23 (msg:"Telnet session detected");

测试:

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#550 (1.484)	TCP flow to low port	2025-05-22 17:22:46	192.168.164.132:48424	192.168.164.131:23	TCP
#551 (1.485)	Telnet session detected	2025-05-22 17:22:45	192.168.164.132:48424	192.168.164.131:23	TCP
#552 (1.486)	TCP flow to low port	2025-05-22 17:22:45	192.168.164.132:48424	192.168.164.131:23	TCP
#553 (1.487)	Telnet session detected	2025-05-22 17:22:45	192.168.164.132:48424	192.168.164.131:23	TCP
#554 (1.488)	TCP flow to low port	2025-05-22 17:22:46	192.168.164.132:48424	192.168.164.131:23	TCP

telnet 192.168.164.131

alert icmp 192.168.164.131/24 any -> 192.168.164.131/24 any (msg:"ICMP PING NMAP"; dsizе:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)

测试:

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#5 (1.338)	ICMP PING NMAP	2025-05-22 17:31:11	192.168.132	192.168.131	ICMP

ping -c 1 -s 0 192.168.164.131

alert tcp any any -> 192.168.164.131/24 21 (content: "USER admin"; msg: "FTP admin login");

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#7 (1.997)	FTP admin login	2025-05-22 17:30:40	192.168.164.132:48756	192.168.164.131:21	TCP

ftp 192.168.164.131

六、思考与心得体会