Computer Science Department
University of Crete

Piggymon: Using Snort IDS for IP Traffic Classification and Throughput Monitoring

Master's Thesis

Peter Politopoulos

September 2011
Heraklion, Greece

University of Crete
Computer Science Department

Piggymon: Using Snort IDS for IP Traffic Classification and Throughput Monitoring

Thesis submitted by
Peter Politopoulos
in partial fulfillment of the requirements for the
Master of Science degree in Computer Science

THESIS APPROVAL

Author: Peter Politopoulos

Committee approvals:

Evangelos P. Markatos
Professor, Thesis Supervisor

Maria Papadopouli
Assistant Professor

Dimitris Nikolopoulos
Associate Professor

Department approval:

Angelos Bilas
Associate Professor, Chairman of Graduate Studies

November 2011
Heraklion, Greece

Abstract

Network traffic monitoring is one of the most valuable tools, utilized by administrators and engineers alike, in order to effectively design, manage and oversee the vast amount of IP networks that comprise the internet. At the same time, Intrusion Detection Systems are widely deployed in an effort to protect the aforementioned networks from attacks ranging from simple worms to sophisticated hacking attempts. Snort is the most widely installed network Intrusion Detection and Prevention System (IDS / IPS) with the added benefit of being free and open-source.

In this thesis we present Piggymon, a passive network monitoring system built on top of an unaltered Snort installation. Piggymon takes advantage of Snort's fast packet classification engine and can categorize traffic based on simple port-matching or more complex, signature-based rules. The traffic monitoring module can run alongside the normal IDS functionality of Snort with minimal performance impact. Besides that, Piggymon is easy to install and customize and will be available as an open-source project after the presentation of this thesis.

Supervisor: Professor Evangelos Markatos

GR

Περίληψη

Η παρακολούθηση της κυκλοφορίας είναι ένα από τα πιο πολύτιμα εργαλεία, που χρησιμοποιείται από τους διαχειριστές δικτύων όσο και τους μηχανικούς, για να αντιμετωπίσουν αποτελεσματικά το σχεδιασμό, τη διαχείριση και την επίβλεψη του τεράστιου αριθμού των IP δικτύων που απαρτίζουν το διαδίκτυο. Ταυτόχρονα, τα συστήματα ανίχνευσης εισβολών διαδίδονται ευρέως , σε μια προσπάθεια να προστατεύσουν τα προαναφερθέντα δίκτυα από επιθέσεις, οι οποίες κυμαίνονται από απλά σκουλήκια εως εξελιγμένες προσπάθειες εισβολής. Το Snort είναι το πιο ευρέως εγκατεστημένο σύστημα ανίχνευσης και πρόληψης εισβολών δικτύου (IDS / IPS), με το επιπρόσθετο πλεονέκτημα ότι είναι δωρεάν και ανοιχτού κώδικα.

Σε αυτή την διατριβή παρουσιάζουμε το Piggymon, ένα παθητικό σύστημα παρακολούθησης δικτύου το οποίο είναι σχεδιασμένο να τρέχει βασιζόμενο σε μια αναλλοίωτη εγκατάσταση του Snort. Το Piggymon εκμεταλλεύεται τη γρήγορη μηχανή ταξινόμησης πακέτων του Snort και κατηγοριοποιεί την κίνηση με βάση είτε απλώς την θύρα επικοινωνίας, είτε πιο σύνθετους κανόνες “υπογραφές”. Η μονάδα παρακολούθησης της κυκλοφορίας μπορεί να τρέχει παράλληλα με την κανονική IDS λειτουργικότητα του Snort με ελάχιστη επίδραση στην απόδοση. Πέρα από αυτό, το Piggymon είναι εύκολο στην εγκατάσταση και την προσαρμογή και θα είναι διαθέσιμο ως εφαρμογή ανοιχτού κώδικα μετά το πέρας της παρουσίασης της διατριβής.

Επόπτης Καθηγητής: Ευάγγελος Μαρκάτος

Ευχαριστίες

Θα ήθελα να ευχαριστήσω βαθύτατα τον επόπτη καθηγητή μου, για την υπομονή και την τεράστια κατανόηση που έδειξε. Οι συμβουλές και οι παρατηρήσεις του πάντα γινόντουσαν με διακριτικότητα, ευγένεια και σοφία.

Επίσης, ευχαριστώ τους συναδέλφους μου στο εργαστήριο κατανεμημένων συστημάτων του Ι.Τ.Ε. και ιδιαίτερα τον Αντώνη Παπαδογιαννάκη για την άψογη συνεργασία κατά την σύντομη παραμονή μου.

Τέλος, η πιο σημαντική συνεισφορά από όλες, αυτή της οικογένειάς μου, που ήταν πάντα παρούσα και μου έδωσε την δύναμη να ολοκληρώσω ό,τι έχω πετύχει μέχρι σήμερα.

Σας ευχαριστώ.

Στην αδελφή μου

Contents

List of Figures

List of Tables

1

Introduction

1.1 The Need for a Flexible and Easy-to-deploy Application/ Host based Network Monitoring System

In 2011 the number of hosts connected to the Internet surpassed the 2 billion [1] mark and still keeps expanding. Along with the growth of host population, the complexity and throughput of the applications that communicate over the network is also on the rise. Huge social networks like Facebook have millions (800 million [32]) of users that exchange media and play flash-based games over the network. Video streaming is nowadays a very common internet activity [29], when only a few years back the throughput required would have been prohibitive. Large clusters serve millions of gamers playing World of Warcraft and other massive multiplayer online games. Amazon S3 is thriving in the distributed server market. It is evident that distributed applications and grid computing are no longer confined in the academic community.

Furthermore, there is an alarming increase in attacks over the internet. Hacker groups are frequently making the headlines for accessing and publicizing classified information [33][34]. Worms can spread rapidly among vulnerable computers, causing havoc. Highly sophisticated distributed denial of service (DDOS) strikes are very difficult to backtrack and stop. Thus, a very large number of Network Intrusion Detection and Prevention Systems like Snort [2], bro [3] and Suricata [4] has been deployed to counter these threats.

Network monitoring provides the information required for a large number of tasks. Some of them are:

- Troubleshooting congestion and subsequent delays.
- Tracking network growth. Demonstrating the need for interlink upgrades, based on verified per-application usage data.
- Designing Quality of Service schemes that meet the specific needs of current usage. Protecting time-critical communications.
- Ensuring that Service Level Agreements are met.
- Proactively alerting of usage problems that would eventually cause legal or technical harm.

Under these conditions, network administrators still have only a handful of handy tools publicly available for categorizing network traffic by the application or the host that generates it. While network traffic classification is an active topic among researchers and highly accurate systems have been developed to meet the demand, most lack the simplicity and the ease of deployment required for a wider adoption.

It is important to note that simple IP port-based classification has been proven insufficient a long time ago [5]. Applications are often configured to use dynamic ports while some others, like Bittorrent [6] and Skype [7], employ sophisticated techniques in order to avoid detection altogether.

1.2 Contributions

In the previous section we have shown that network monitoring is an essential tool needed by professionals and engineers alike. On the other hand, Intrusion Detection and Prevention Systems (IDS/IPS) are already installed and running on a variety of networks worldwide. Thus, the objective of our work is to develop a system that provides application and/or host throughput information utilizing existing IDS/IPS, without altering their code. The design of our solution must be such that it will not slow down the original intrusion detection functionality, while retaining as much of the accuracy and flexibility of a standalone monitoring system as possible.

The contribution of this thesis is the introduction of Piggymon. Piggymon is a network monitoring system built on top of a totally unaltered Snort installation. The system allows for per-host and per-application throughput monitoring. In addition to that, application traffic classification can be based not only on port number but also on signature detection. Users can view statistics in text-mode or rrd [8] generated graphs. The solution provided is easy to install and fast enough to run even on embedded devices.

1.3 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 presents the implementation architecture of Piggymon, illustrates some examples of its usage and discusses the main features that it offers. In Chapter 3 we demonstrate some real-world examples of Piggymon usage, we evaluate its performance and compare the application detection mechanism with existing solutions. Chapter 4 gives a synopsis of the related work on the area of network traffic classification. Finally, Chapter 5 concludes the thesis and presents possible improvements and future work.

2

System Design

2.1 Piggymon Design

A Network Intrusion Detection System (NIDS) is a system that tries to detect malicious activity such as denial of service attacks, port scans or even attempts to hack into computer systems by monitoring network traffic. Snort is a widely deployed open source network intrusion prevention and detection system. It uses a rule-based language combining signature, protocol and anomaly inspection methods.

Piggymon is build on top of Snort NIDS and consists of a set of rules and the application executable. Thus, some of the underlying functionality of Snort must be discussed, for the shake of explaining the architecture of Piggymon thoroughly.

2.1.1 Snort detection engine, rules and flowbits

Figure 2.1 depicts how Snort is structured. Frames are initially received from a capture library. Older versions supported only lib-pcap [9] but more recent ones (2.9+) can acquire frames from a variety of capture interfaces through the Data Acquisition facility, or DAQ for shorter. Once a frame is received it is first handed to the packet decoder. The decoder takes frames from different types of network interfaces (Ethernet, SLIP, PPP…) and prepares them for processing by removing the link-layer dependent encapsulation. Next, the decapsulated packet is received by the preprocessors. Of the many Snort built-in preprocessors, Piggymon depends only on two: frag3 and stream5 [10].

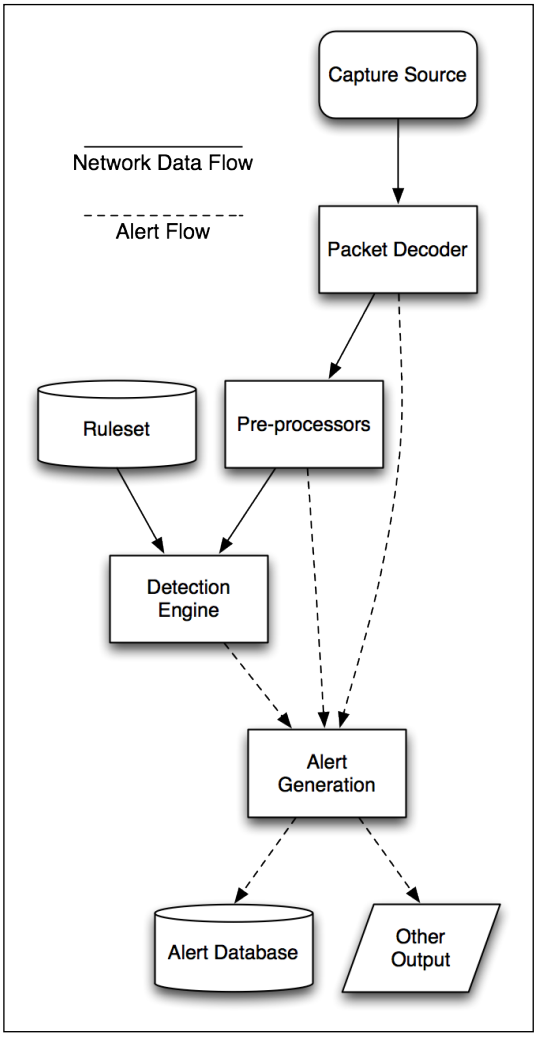

Figure 2.1 Basic Snort Architecture

Frag3 reassembles the packet fragments into the original packet, handling retransmissions and overlaps between them. Packet defragmentation methods differ among hosts, allowing an application that successfully exploits these differences to avoid detection. Although this was originally noted as an important weakness in Intrusion Detection Systems [11] it can be employed for application detection avoidance as well. In an even simpler fashion, plain string matching algorithms used by some traffic classification engines will fail when the search string is divided among packet fragments. For example, if the traffic classification engine looks for the string "***GNUTELLA CONNECT***" the application client can divide it among two overlapping frames containing "***GNUTELLA CO***" and "***A CONNECT***" and escape identification.

Stream5 handles the reassembly of higher-level Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) sessions. It is essential at this point to clarify the meaning of the term "network flow" or "flow". From this section and on a flow will be defined as the data stream contained in the packets exchanged between two hosts, uniquely identified by the source and destination IP addresses, as well as the corresponding source and destination ports. TCP flows are initiated by the classic 3-way TCP handshake, while

UDP network flows are established as the result of a series of UDP packets from two end points via the same set of ports. Stream5 additionally tracks Internet Control Message Protocol (ICMP) messages for the purposes of checking for unreachable and service unavailable messages, which effectively terminate a TCP or UDP session.

Piggymon relies on a feature of stream5 called “flowbits”. Flowbits are 1-bit variables that apply to network flows. Once a flowbit is set for a specific flow, all packets belonging to that flow retain the set (‘1’ or true) value.

Both frag3 and stream5 are target-based, meaning that if properly configured they emulate accurately the behavior of the target host, ‘viewing’ the data stream in exactly the same way. Besides that, both preprocessors are enabled by default in a Snort installation and are so vital in the functionality of the IDS, that it is very rare for an administrator to disable them.

2.1.2 Snort Detection Engine and Rules

Snort’s main detection functionality relies on rules. A Snort rule consists of two major and parts as shown on Figure 2.2. Our example consists of a rule that matches bittorrent Distributed Hash Table (DHT) protocol ping messages.

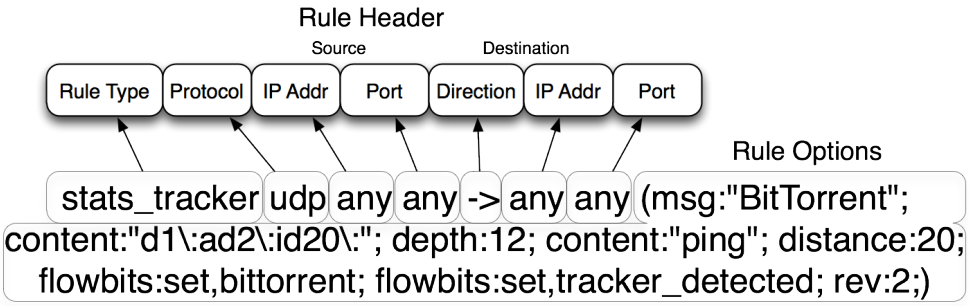

Figure 2.2 Anatomy of a Snort Rule

The rule header begins with the rule type. Common rule types are alert, log and drop. Rule types define the action that Snort should take in case a packet matches the specific rule. Stats_tracker is a custom type, designed for use with Piggymon. We will explain all custom rule types that we introduced in section 2.1.4. The rule header includes protocol (IP, TCP or UDP) as well as source and destination IP addresses and ports. Between the source and destination, an arrow (-> or <-) denotes the direction that the packet must have in order to match the rule. In this example directionality is of no importance, since source and destination can be any IP address on any port, as long as the packet is of the UDP type.

The rule options part, is where the more complex aspects of detection reside. The first option in the example is the message that Snort will display in case of a match. Following that, two content matching options instruct Snort to search

for the specific DHT ping strings. The first string can only be found at the start of the packet, as it is 12 bytes long and we limit the depth of the search to 12 bytes. The second string should be at least 20 bytes further, after the first. Snort uses by default the Aho–Corasick [12] string matching algorithm which while it is fairly fast, its performance may vary widely depending on the pattern options and on network traffic variance [13]. This performance variance can be narrowed with careful rule construction. In the example given, the second string search for "ping" is not performed in case the first one, which is longer and infrequent, fails. If both strings are found in a UDP packet, the rule instructs Snort to set two flowbits. The last option is the rule revision, used for versioning purposes and does not affect the detection engine.

2.1.3 Snort Alert Export Mechanism

Snort supports various ways of exporting alerts depending on the selected output module. Output modules were introduced in version 1.6 and include export capabilities for syslog, text files, unix sockets, databases, unified logs and CSV files. Csv stands for Comma Separated Value files, where each line represents a matching packet and the attributes that are selected for logging are separated by commas. Each rule type can use independently any of the available output modules.

2.1.4 Piggymon Architecture

While Snort allows the inclusion of custom pre-processors and dynamic rules [14] that would greatly ease the design of Piggymon, these features are not used based on the fact that they are only compatible with the specific version of Snort that they are compiled against. That would severely limit the ability to use Piggymon on any already present Snort installation and would beat the purpose of the thesis.

Therefore, the features of Snort that are essential for Piggymon to run are rule-types, rules, flowbits (stream5) and CSV file export. These features are present and functioning in nearly every deployed version of Snort since its establishment.

With the inclusion of Piggymon.conf file, we introduce the following configuration changes:

- **Configuration of the event-queue**

    config event_queue: max_queue 1 log 1 order_events priority

This line instructs Snort to log only one event per packet received. In the case multiple rules match the packet, the one with the highest priority will only

produce an alert. The event ordering by priority option is unfortunately bugged until very recent versions of Snort. Piggymon gains several advantages if it is run on a recent, non-bugged version. For instance, this would allow packets that generate IDS alerts to be correctly accounted for throughput statistics as well.

- **Definition of new rule-types**

Several new rule-types are introduced. Their names are stats, stats_ports, stats_tracker_detected, stats_tracker and stats_undetected. They are defined as follows:

```
ruletype stats_undetected
{
  type alert
  output alert_CSV: /var/log/Snort/stats/Piggymon.log msg,dgmlen,srcport,dstport
10M
  output log_null
}
```

All stats rule-types are of the subtype alert. They all also output their alert logs to a CSV file. Each line will include the alert message, the packet length and (optionally for stats_undetected) source and destination port. The inclusion of source and destination ports on stats_undetected aid the discovery of popular undetected ports in the traffic. The log-files that these alerts produce are parsed, aggregated, rotated and sorted by Piggymon. Finally the log_null instruction commands Snort to not log the actual packet data for stats_* type alerts, since they are not needed and it would be stressful for the hardware.

- **(Optional) Disable checksumming**

Piggymon can be used for categorizing network traffic contained in trace files. Most network traffic dumpers capture the first few bytes of each packet, making the Cyclic Redundancy Check (CRC) check invalid. Moreover, modern or specialized Network Interface Cards frequently offload the computation of checksums onto the card itself, again causing a checksum mismatch. In both cases, Snort by default will discard the packet. To avoid this behavior we include the following line:

```
config checksum_mode: none
```

- **Inclusion of Piggymon rule definition files**

The last part of the Piggymon.conf is the inclusion of the files that contain the rules for traffic classification.

```
include $RULE_PATH/stats_ports.rules
include $RULE_PATH/stats_high_popular.rules
include $RULE_PATH/stats_multi.rules
```

include $RULE_PATH/stats_tracker_detected.rules
include $RULE_PATH/stats_tracker.rules
include $RULE_PATH/stats_undetected.rules

Snort stores internally a rule application order. The sequence of the definitions is very important. After parsing the configuration files the rule application order will be:

(Former rules, alerts, logs etc) ➡ (stats_ports) ➡ (stats_high_popular) ➡ (stats_multi) ➡ (stats_tracker_detected) ➡ (stats_tracker) ➡ (stats_undetected)

We already instructed Snort to produce at most one alert for each packet, so in its way along those rule types it will be logged only once. If it is not detected as an actual attack from the IDS it will be passed in turn to Piggymon stats rules. The way rule files are prioritized is the basis for optimum traffic classification. The actual flow of a packet detection is depicted in figure 2.3.

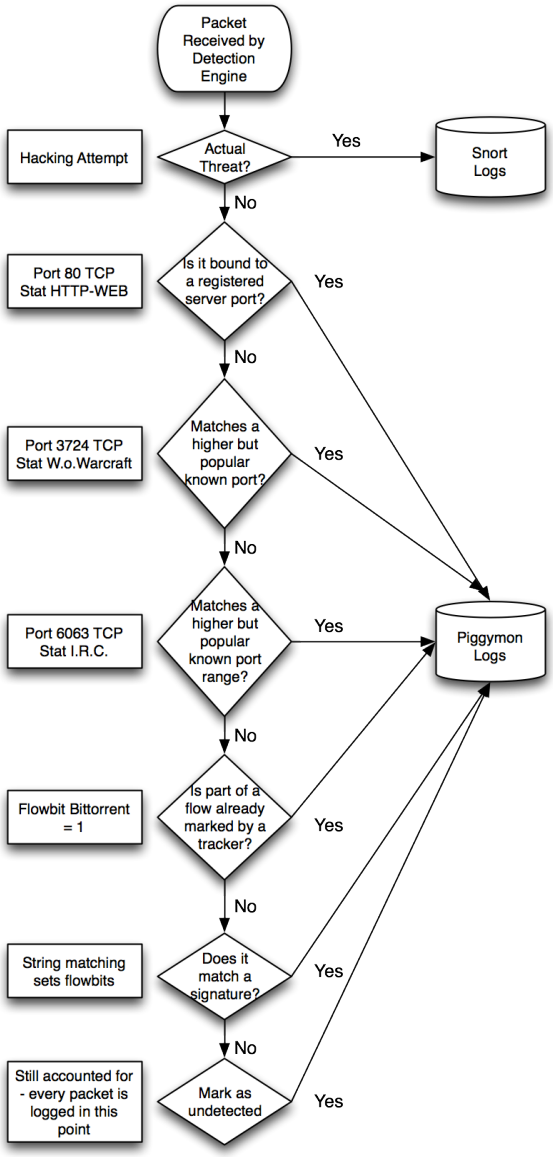

Figure 2.3 Packet Classification Flow

Piggymon rules initially do a fast check on the TCP or UDP port number. This number is first compared to registered, known low ports and continues with higher and less frequently used numbers or ranges. The check is rapid, since it is done by an efficient port group data structure inside Snort. If there is no port match, the packet continues to the tracker_detected rules. The function of these rules is to check whether a flowbit is set by a tracker signature match, earlier in the same flow. If there was no flowbit set, the tracker rules begin the costly in computational terms, pattern matching operations. Therefore, we use the tracker_detected rules as a way to perform searches only until the first recognition of the application generating the network flow. If all rules fail, the packet is tagged as uncategorized traffic but still written in the Piggymon log.

ICMP packets are accounted for, as a separate traffic category.

2.2 Port-based detection rules

There are three individual port rule files included with Piggymon. The first port rule file (stats_ports.rules) contains the definitions for well-known port numbers, in the low range of 1 to 1023, tcp or udp. The overwhelming majority of the rules is compiled from the official list maintained by Internet Assigned Numbers Authority (IANA) [15]. The second file (stats_high_popular.rules) is a collection of the most commonly used higher ports combining official and unofficial sources, such as IANA, SANS Institute Internet Storm Center [16] and SpeedGuide.net [17]. The last file (stats_multi.rules) contains protocols that use ranges of ports for communication. Splitting the port rules files allows the user to easily disable any of them, in order to achieve the desired performance and accuracy that meet the specific installation needs. More detailed discussion of customization is presented on section 2.6, ***Usage Examples***.

The following rule is part of the first file and categorizes Telnet traffic:

stats_ports tcp any any <> any 23 (msg:"Telnet"; flowbits:set,port_detected; sid: 2000028; rev:1;)

Piggymon distribution comes with a helper tool designed to convert comma separated value (CSV) files into Snort rules. The executable is located into the ***tools*** directory under the name ***rule_creator***. Various sources provide port data in CSV format or other similar table formats that can be easily converted. Rule_creator accepts a single parameter, defining the starting signature id, reads the port CSV from standard input and generates the rules in standard output. The example rule shown above was generated by a CSV input of:

23,TCP,,Telnet

For stats_ports.rules, the Signature ID (SID) range begins at 2.000.000 and the example rule is the 28th from start. The creators of Snort recommend the use of SIDs of more than 1.000.000 for custom user rules. SID numbering could

be omitted, but this would cause warnings in several Snort installations and greatly inhibit rule performance profiling.

2.3 Signature-based detection rules and Unknown traffic

The signature-based detection rules are the most computationally costly components of Piggymon and are split into two sub-groups. Rules inside stats_tracker.rules detect applications by matching string patterns or more-complex protocol signatures. In case a match is found, the entire flow of the corresponding packet is marked by setting a flowbit. All the following packets belonging to the marked flow are then tracked by the second sub-group of rules that reside in stats_tracker_detected.rules file. The group we present here tracks BitTorrent protocol traffic.

Inside file stats_tracker_detected.rules

stats_tracker_detected ip any any <> any any (msg:"BitTorrent"; sid:1100001; flowbits:isset,bittorrent; rev:1;)

Inside file stats_tracker.rules

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"GET /scrape?info_hash="; depth:100; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909003; rev:1;)

...

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"d1|3A|rd2|3A|id20|3A|"; depth:20; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909014; rev:1;)

"Tracker detected" rules are checked ***before*** "tracker" rules by definition order inside Piggymon.conf (Sub-Section 2.1.4). Therefore, once a flow is marked, packets no longer undergo the expensive "tracker" rule checks. Finally, everything that is not matched by port nor tracker rules is reported as unknown type of traffic.

Inside file stats_undetected.rules

stats_undetected tcp any any <> any any (msg:"UNKNOWN TCP"; sid:1300000; flowbits:isnotset,port_detected; flowbits:isnotset,tracker_detected; rev:1;)

stats_undetected udp any any <> any any (msg:"UNKNOWN UDP"; sid:1300001; flowbits:isnotset,port_detected; flowbits:isnotset,tracker_detected; rev:1;)

The above catch-all rules are applied always last in order. The flow bit port_detected is set by port-matching rules with the purpose of not classifying data as both known and unknown, in installations of Snort that allow multiple alerts per packet.

2.4 Host-based traffic classification

Configuring Piggymon for host-based classification is fairly straightforward. For each of the monitored hosts or subnets, a rule line should be added inside stats.rules:

stats ip MY_HOSTS any <> any any (msg:"MY_HOSTS_NAME";)

Where MY_HOSTS is the subnet (or single host) being monitored and MY_HOSTS_NAME is the label displayed by Piggymon and rrd for the matching traffic. Under the default configuration, traffic reported by host-based rules is not checked against application detection (ports/tracker) rules.

2.5 A note on directionality

In many cases, information on the direction of the traffic is required. For instance, a network administrator may need to differentiate between web traffic belonging to internal servers and web traffic served by external servers to local clients. In these cases, it is essential to define what hosts belong to the internal network. This is achieved by setting the value of $HOME_NET variable inside Snort.conf. The next step depends on the type of classification being applied. For port-based detection, Piggymon requires four rules per known port.

stats_ports tcp any any <> any 23 (msg:"Telnet"; ... ;)

is substituted by:

stats_ports tcp $HOME_NET any -> any 23 (msg:"Telnet OUT"; ... ;)
stats_ports tcp $HOME_NET 23 -> any any (msg:"Telnet OUT"; ... ;)
stats_ports tcp $HOME_NET any <- any 23 (msg:"Telnet IN"; ... ;)
stats_ports tcp $HOME_NET 23 <- any any (msg:"Telnet IN"; ... ;)

In the case of host-based detection rules, their number will have to double

stats ip MY_HOSTS any <> any any (msg:"MY_HOSTS_NAME";)

is substituted by:

stats ip MY_HOSTS any -> any any (msg:"MY_HOSTS_NAME OUT";)
stats ip MY_HOSTS any <- any any (msg:"MY_HOSTS_NAME IN";)

In a similar fashion unknown types of traffic can show direction as well

stats_undetected tcp any any <> any any (msg:"UNKNOWN TCP"; ... ;)

is substituted by:

stats_undetected tcp $HOME_NET any -> any any (msg:"UNKNOWN TCP OUT"; ... ;)

stats_undetected tcp $HOME_NET any <- any any (msg:"UNKNOWN TCP IN"; ... ;)

2.6 Configuration examples

This section lists a few example environments suitable for Piggymon deployment. Along with each example, a proposed configuration is described. This list cannot cover every possible configuration and therefore it serves the purpose of a simple guide.

2.6.1 Existing IDS installations, heavy traffic

In cases where Snort processes traffic at rates close to the hardware limits, Piggymon should be configured to run as light as possible. Slow, pattern-matching trackers should be either disabled or reduced to a small, customized subset of the original rule group. This subset can be adjusted to include only rules matching the most prominent application protocols of the specific network segment. Of course, this selection should not be permanent since usually over the years, network usage patterns vary greatly.

It would be advisable to store the Piggymon log files on a separate physical storage device than Snort log files. Ideally, the system would have more than one processing core, running Piggymon and Snort on different ones. Recent versions of Snort would be preferable, since they provide significant performance improvements.

2.6.2 Small office / Residential monitoring

Running Piggymon in a Small Office / Home Office (SOHO) environment presents many advantages. Usually the overall throughput is relatively low, in the range of a few Megabytes per second. Consequently, Snort and Piggymon can be run on the router providing continuous per-application throughput information as well as IDS/IPS functionality. Piggymon was successfully installed on a router running the Open-WRT firmware [18] and provides statistics for a residence with 5 computers.

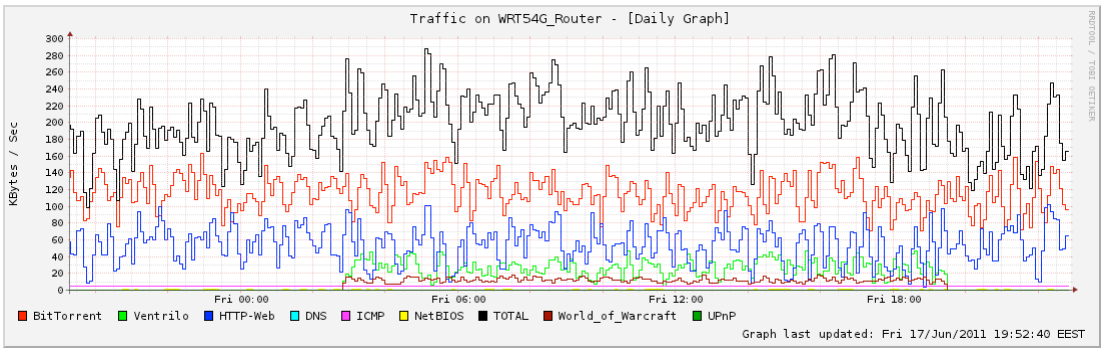

Figure 2.4 Piggymon traffic graph on a SOHO environment

An additional advantage of SOHO installations is that the vast majority of such networks operate under Network Address Translation (NAT). Usually the

translated, internal addresses are not routable and therefore not visible to the external world. In order for an internal host to accept a connection from the internet, the administrator has to manually add a static entry in the translation table of the router that includes the internal host IP address and the listening application port number. This procedure also requires the listening (server or peer to peer) applications to be run on a static, known port. Monitoring such applications is as easy as adding the corresponding port rule entry for Piggymon, in case it's not already there. For more modern routers that support dynamic IP/port allocation by Universal Plug'n'Play (UPnP) messages, a daemon would be required, that would listen for the messages and add the appropriate rules, restarting the Snort engine after each new allocation.

In the SOHO usage example, Snort would be configured with a light set of rules and Piggymon should have most of the tracker rules disabled, due to processing power constraints. The system could be run on the DSL or Cable router, a device that is always on and consumes little energy. Another possible SOHO application would be the monitoring of each of the individual hosts in the Local Area Network (LAN), in order to troubleshoot delays or account for per-host bandwidth consumption.

2.6.3 Web hosting services

Snort can differentiate between web traffic belonging to different domains, by using tracker rules and flowbits in a similar fashion to the other tracked protocols. This allows Piggymon, with the appropriate rule set, to track and graph bandwidth usage by domain or by client.

An example implementation, is presented below:

Inside file stats_tracker_detected.rules

```
stats_tracker_detected tcp $HOME_NET 80 <> any any (msg:"Client X";
flowbits:isset,client_x; sid:1100000; rev:1;)
```

Inside file stats_tracker.rules

```
stats_tracker tcp $HOME_NET 80 <> any any  (msg:"Client X"; flow:
established,to_server; content: "ClientXBlog.com"; fast_pattern; nocase; http_header;
pcre:"/^Host\x3a\s*[a-z0-0\.-]+\.ClientXBlog.com/smi"; http_header;
flowbits:set,client_x; sid: 1900000; rev:1;)

stats_tracker tcp $HOME_NET 80 <> any any  (msg:"Client X"; flow:
established,to_server; content: "ClientXCorp.com"; fast_pattern; nocase; http_header;
pcre:"/^Host\x3a\s*[a-z0-0\.-]+\.ClientXCorp.com/smi"; http_header;
flowbits:set,client_x; sid: 1900001; rev:1;)
```

Each Hypertext Transfer Protocol (HTTP) session contains a request Uniform Resource Locator or Universal Resource Locator (URL) that concludes with the parent domain name. The above tracker rules detect the presence of the entirety of this name, to avoid false positives. The pcre option instructs Snort

to use LibPCRE [22] regular expressions for matching the unknown, dynamic part of the name inside the packet, as well as the static one. Since this search is relatively slow, the content match option restricts the packets inspected to only the ones that already at least contain the partial (parent) name. In addition to that, a further optimization is achieved by the tracker rule pair that excludes from searches packets belonging to an already-classified flow.

A similar rule set can provide a large number of useful functionalities. For example, a company network administrator can track specific site usage, like e-bay or Facebook, as long as it takes place inside the corporate network.

3

Experimental Evaluation

3.1 Experiments Setup

Most of the experiments conducted were based on captured traces. Network trace files allow for reproducible and comparable result collection. There are a total of four files presented in this thesis. Two of the files originate from the connection between the University of Crete [19], Knossos campus and GrNet network (and consecutively the internet) [20]. The data on the files contain packets flowing in both directions, starting at the midday of July, 25th in 2006. Each file contains a few minutes of traffic, reaching a file-size of 2 Gigabytes and a total of 4 Gigabytes.

An additional two trace files come from a residential installation of Piggymon. One of them is mainly consisted of bittorrent traffic, while the other one represents normal multi-hour usage by the residents. Their size is in the scale of hundreds of Megabytes.

All trace files are captured by tcpdump [9] with the maximum capture length (snaplen) left to the default value of 68 bytes for IPv4 packets. The host running the tests was provided by the Distributed Computing Systems Laboratory of the Foundation for Research and Technology, Crete, Greece [21]. The detailed technical specifications of the host are presented on Table 3.1.

	Test System Specifications
CPU	Intel® Xeon™ 2.40GHz, 512KB Cache
Memory	2 GB
Operating System	Debian GNU/Linux, kernel 2.6.32-5-486
Disk Drive	40GB ATA
Snort Version	2.9.1 IPv6 GRE (Build 71) - libpcap 1.2.0

Table 3.1 Test Host Specifications

3.2 Measurement Accuracy

Piggymon in most Snort installations (prior to 2.9.1) cannot account for the traffic that the actual IDS logs. Usually, this is a very small portion of all traffic and in all our tests it was under 0,5% of actual data. Besides that, if the IDS produces a large amount of alerts, it is a sign for the Network Administrator to take action and fix the reported problems. In case the alerts are issued by harmless traffic, it is recommended to disable the chatty rules that flood the alert facility.

Furthermore, Piggymon cannot process packets that are not supported by the specific installation of Snort. For example, IP version 6 support requires a version higher than 2.8.4 compiled with the " --enable-ipv6" flag set.

Finally, the preprocessors sometimes create virtual packets that are checked by the detection engine and consequently Piggymon. In section 2.1.1 we discussed data stream recreation and why it is imperative for the IDS and Piggymon to view the traffic exactly the same way as the communicating hosts. It is possible to disable the stream-inserted packets by uncommenting the following line inside Piggymon.conf:

#config detection: no_stream_inserts

This should be only done with the purpose of maximizing Piggymon packet or byte accounting precision and it is not recommended in actual production systems. In table 3.2 we present the packet and byte counts of the four trace files used throughout the experiments in the thesis.

	Actual Packets	Actual Bytes	Piggymon Packets	Piggymon Bytes	Difference % - Packets	Difference % - Bytes
UoC 1	2775381	1955594261	2811121	2072234334	1,29%	5,96%
UoC 2	2734972	1956241763	2769392	2068828470	1,26%	5,76%
residential	699788	130722263	709525	133042059	1,39%	1,77%
residential with torrent	107526	77758506	108549	78079509	0,95%	0,41%

Table 3.2 Piggymon packet and byte count deviation

Differences in packet counts were in every test, including countless not presented here, less than 1,4% of the total. Byte counts were significantly less accurate reaching differences of up to 6%. Despite the fact that 6% is not negligible, it is evident by the numbers presented in the following experiments that virtual packets introduced by Snort are either part of unclassified traffic or distributed equally among protocols, thus preserving the correct traffic ratio. In order to strengthen this point, all the following experiments are conducted with the stream-inserted packets option enabled.

3.3 Comparison with Other Network Monitoring Tools

For the classification comparison we will introduce the reader into three popular and free classifications systems. Each one of the systems can take tcpdump trace files as input and generally use libpcap for processing.

3.3.1 OpenDPI

OpenDPI is the open source version of ipoque corporation's deep packet inspection engine [23]. OpenDPI is a software library implementing various classification methods, including pattern matching, behavioral and statistical analysis. The authors claim that the library is capable of reliably detecting protocols and applications in the network, even when they are proprietary, encrypted or obfuscated.

In order to conduct the experiments below, the latest stable version (1,3) was installed on the test system. Each trace file was subsequently processed by the demonstration application included with the library.

3.3.2 Tie & Tie Stats

Tie is a novel, community-oriented software for traffic classification, which aims at becoming a common tool for the fair evaluation and comparison of different characterization techniques [24]. Tie fosters the sharing of common implementations and data. This is achieved by adapting a modular design, that allows the use of various classification engines and statistics aggregation mechanisms.

Unfortunately, the architecture of Piggymon and its dependance on Snort's framework does not allow for interoperability with tie. The experiments conducted utilized the classifiers that are included in the latest distribution of tie (version 1,0,0 beta 3).

3.3.3 Appmon

Appmon is a passive monitoring application for per-application network traffic classification [25]. Appmon uses deep packet inspection to accurately attribute traffic flows to the applications that generate them, and manages to classify live traffic up to Gigabit speeds. Appmon is deployed globally as LOBSTER's [26] main sensor technology.

Since appmon is oriented towards live traffic monitoring and graphing, a specialized version had to be used in order to obtain directly comparable results from trace files. This version is named appmon standalone, or appmon-sa and allows the use of tcp dump files as input.

3.3.4 Resource Consumption

The first step in performance evaluation is a direct comparison between execution times for each of the trace files. Table 3.3 presents the results obtained by means of the unix ***time*** command. The only exception to that rule is the fourth column, representing the runtime of Snort with Piggymon rules. On this specific occasion, the true running time is reported by Snort instead of ***time*** and does not include the constant initialization period needed by the Snort engine. Each execution time listed, represents a mean value obtained by 10 consecutive runs. The last column labeled "Piggymon log parsing" lists the time needed by Piggymon in order to process Snort output logs and produce sorted, text or graphical output of the classification process. Appmon and OpenDPI achieve the same result in a one-step process. The interval elapsed for tie includes the runtime of tie-stats, that produces human-readable results of the classification result and is generally less than a second, even for large trace files.

	appmon	opendpi	tie	Snort with Piggymon rules	Piggymon log parsing
UoC 1	43s 670ms	6m 54s 680ms	44s 890ms	1m 44s 880ms	23s 540ms
UoC 2	22s 820ms	5m 36s 950ms	5s 800ms	1m 34s 20ms	22s 300ms
residential	1s 240ms	1m 1s 960ms	0s 990ms	22s 160ms	5s 210ms
residential with torrent	1s 630ms	3s 460ms	0s 239ms	5s 470ms	6s 40ms

Table 3.3 Comparison of execution times

Appmon and tie are the fastest classifiers. This was expected since they are designed and optimized for the specific purpose. OpenDPI was exceptionally slow, especially when dealing with multi-gigabyte files. The authors claim that this is not the case for the commercial version which is significantly faster.

In a real-world scenario Snort and Piggymon would run in parallel, taking advantage of multi-cpu and multi-storage systems. Moreover, our initial assumptions were that Snort would already be running on the monitoring system, with the addition of Piggymon rules. The Piggymon executable would be run as a cron job every 5 minutes, parsing and rotating the relevant logs. Consequently, a more useful experiment is presented in Table 3.4, where Snort performance is compared between various rule sets. For this experiment, both of the University of Crete trace files were used in 10 consecutive runs and the average is listed. Megabits per seconds and packets per second are reported by Snort's performance monitor (perfmon) plugin. The first column represents the processing speed of Snort on the test system, with only the latest official (July 2011) sourcefire ruleset enabled. The second column is the performance with only the default Piggymon rules enabled, while the third lists the rates with

both sourcefire and Piggymon rules enabled. Without the time-consuming tracking rules enabled, Piggymon has a minimal impact (≋ 6%) on Snort performance. On the other hand, even with tracking rules enabled, Snort retains a processing speed above 100 Mbit / sec even on the relatively outdated test system.

		Trackers Disabled		Trackers Enabled	
	sf VRT rules	Piggymon rules	both	Piggymon rules	both
MBit / sec (wire)	202,112	219,281	189,693	163,476	112,693
Packets /sec (wire)	35835	37937	32823	29100	19488

Table 3.4 Slowdown of Snort by Piggymon Rules

3.3.4 Classification

Before proceeding with the results, it is important to note that there are several factors inherent to the use of live traces that severely limit the detection capability of all the tested systems. There is no current "ground truth" classification available for live traces, nor it is possible to create one. Synthetic traffic can be tailored to match (or not match) in it's entity any classification system, producing biased results. This is a well-known issue in the research community and discussed in section 3.4. Moreover, the snap length restriction, prohibits classifiers from using information that may have been available deeper in the packet. Also, flows that begun before the start of the trace capturing but spanned in its duration are usually not classified, as the negotiation strings indicative of the protocol, are usually exchanged at the start of the communication. All these factors contribute in the classification of a large amount of packets as unknown types by all the tested systems, especially in the case of the large University of Crete traces.

Table 3.5 presents the classification result from all the tested systems, for the residential trace containing mostly bittorrent traffic and some web "surfing". Appmon and Piggymon are the only classifiers correctly identifying most of the peer-to-peer traffic on this test. The Domain Name System (DNS) and NetBIOS classes are indicative of the accuracy Piggymon can achieve, despite the fact that the stream inserted packets are enabled. The ruleset used was the full default Piggymon ruleset, including trackers.

Piggymon	Percent	Packets	Bytes	appmon	Percent	Packets	Bytes
HTTP	8,79%	10165	6835666	HTTP	8,84%	10390	6876711
DNS	0,04%	229	34045	DNS	0,04%	229	34045
NETBIOS	0,13%	873	102739	NETBIOS	0,13%	873	102739
BitTorrent	90,26%	91635	70185759	BitTorrent	90,30%	91804	70221693
eDonkey	0,00%	0	0	eDonkey	0%	2	184
Unknown	0,70%	3901	544369	Unknown	0,67%	4228	523134
OpenDPI	**Percent**	**Packets**	**Bytes**	**tie**	**Percent**	**Packets**	**Bytes**
HTTP	8,71%	9502	6769854	HTTP	8,05%	9983	6259846
DNS	0,04%	229	34045	DNS	0,03%	229	24019
NETBIOS	0,13%	805	97827	NETBIOS	0,02%	384	18402
BitTorrent	0,06%	298	49695	BitTorrent	0,01%	66	5272
eDonkey	0,00%	0	0	eDonkey	0,00%	0	0
Unknown	91,06%	96611	70803145	Unknown	85,24%	96027	66278890

Table 3.5 Traffic classification comparison for residential trace file

In Table 3.6 we present the classification result for a number of major indicative classes of traffic inside the University of Crete trace files. Each system presented a large number of traffic classes making the full listing impractically large for inclusion in this thesis.

	Piggymon		appmon		OpenDPI		tie	
	Bytes	%	Bytes	%	Bytes	%	Bytes	%
HTTP	194689772	9,96%	183640297	9,39%	92105405	4,71%	112725588	5,76%
SSH	75908	0,00%	180518	0,01%	Not Tracked	0,00%	12800	0,00%
SMTP	10564259	0,54%	10304781	0,53%	9735970	0,50%	9326165	0,48%
DNS	3793038	0,19%	3793038	0,19%	3463481	0,18%	2371191	0,12%
NetBIOS	3554147	0,18%	3559471	0,18%	27220	0,00%	2990093	0,15%
Warcraft	276040	0,01%	271118	0,01%	Not Tracked	0,00%	786	0,00%
BitTorrent	522987270	26,74%	117280982	6,00%	65362903	3,34%	54595056	2,79%
DC++	14669171	0,75%	11901574	0,61%	8772333	0,45%	188096	0,01%
Unknown	1011424245	51,72%	1479234652	75,64%	1699478648	86,90%	1493309875	76,36%

Table 3.6 Traffic classification comparison for University of Crete trace file

There is a major difference between bittorrent traffic detected by Piggymon and the rest of the classifiers. This is the result of the careful fine-tuning of Piggymon, in order to correctly detect bittorrent messages. In other words, all of the systems would report similar results if the same rules were translated and applied to them. This is discussed in greater detail in the next section.

3.4 Piggymon Detection Capabilities

An objective way to evaluate Piggymon capabilities could not include rules and live traffic. A biased experimenter may adapt the rules to the specific tests, achieving complete, or nearly-complete classification. On the other hand, the use of identical detection rules would only present differences in performance, as the classification results would be identical as well, among every classification system.

The only remaining way to compare classification systems, is direct and based on detection engine capabilities. Piggymon can detect applications that can be distinguished by port numbers, host IP, string matching and protocol "dialogues" based on any combination of the above. Appmon, OpenDPI and tie can match and even surpass all piggymon detection capabilities, given the appropriate tracker / rule set. The main advantage of piggymon is the ease of adding a new classification rule.

Piggymon doesn't offer statistical analysis of host behavior as classification option, because Snort doesn't. This kind of analysis has proven successful by systems such as KISS [27]. Piggymon also does not allow for side-channel analysis, that proves to be a very important tool. For example, suppose an File Transfer Protocol (FTP) transfer is initiated inside an FTP flow, that can be detected by Piggymon. After the negotiation completes, the actual transfer takes place on a separate flow, that is not detected by Piggymon. Side-channel analysis support, would allow a classifier to parse and "remember" the negotiation part, including the information needed (IP, Ports) in order to recognize the spawned transfer flow.

Unfortunately, these techniques prove less and less useful, as encryption is applied on communication channels and protocols such as FTP are being substituted by S(ecure)FTP. In general, Piggymon cannot detect obfuscated and heavily encrypted protocols that use dynamically negotiated ports, such as Protocol encryption (PE) or message stream encryption (MSE) [28] used by the majority of modern bittorrent clients. On the other hand, there is no system at this time capable of detecting and classifying such protocols accurately. This actuality, combined with the fact that peer to peer usage is on the decline [29] possibly mean that these shortcomings of Piggymon are of declining importance.

4

Related Work

Network traffic classification and intrusion detection are both widely researched by the scientific community. So far, their combination had the sole goal of reducing IDS loads, by excluding traffic characterized as harmless. In their paper, Z. Zhang et. al. proposed HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification [31] with this goal in mind. We already mentioned Bro [3] IDS in previous sections. Bro, does traffic classification as part of the intrusion detection process, but there is no known way of extracting the results in a meaningful way for a network administration so far.

Furthermore, the impact of peer-to-peer traffic on intrusion detection has been found to be of major importance [30]. Many of the practices of file sharing programs are mislabeled as attacks. Piggymon can overcome this problem, by careful selection of the detection rules, for both benign peer-to-peer traffic and malicious attacks.

To the best of the writer knowledge there has been no immediate use of an intrusion detection system for direct traffic classification and statistics extraction until today.

5

Conclusions and Future Work

Piggymon is not the fastest nor the most precise network monitoring system available. It can be installed and customized relatively easily. Furthermore, the use of an un-altered Snort IDS engine, allows for the rapid deployment as well as regular improvements (bug-fixes, new features, optimizations) by the open source community, without specifically targeting Piggymon.

Many improvements are planned for Piggymon. Most of them can be contributed by users that adapt it to their specific needs. Every signature supported by the Snort engine can be translated into Piggymon rules. Universal Plug'n'Play support would allow for dynamic application tracking. OpenWRT support has plenty of space for performance and graphing improvements. Nearly all systems capable of running Snort could be readily supported. Additional live installations would reveal bugs and inefficiencies that are yet to be discovered. More options could be coded for different sorting and counts. Lastly, we shall produce a method for efficient, in-Snort side-channel analysis for the tacking of some harder-to-detect protocols.

Appendix A

Piggymon Installation

The installation instructions in this section assume that a Snort installation is already present on a UNIX system. Microsoft Windows and OpenWRT installation is similar, but needs extra steps for the rrd export functionality to be present.

For debian and debian-based operating systems like Ubuntu, there is an automated install script available inside the Piggymon distribution file, named Piggymon-install.

Step 1. Decompress Piggymon files

On the directory where you downloaded the latest Piggymon files type:

tar xvzf Piggymon.tgz

Step 2. Install rule files

Copy the rule files into the Snort rule file directory (usually /etc/Snort/rules/):

cp ./Piggymon/rules/* /etc/Snort/rules

Step 3. Create the stat logging directory

Create a directory for Snort to log packet information

(default /var/log/Snort/stats):

sudo mkdir /var/log/Snort/stats

sudo mkdir /var/log/Snort/stats/rrds

chown Snort /var/log/Snort/stats

Step 4. Include the Piggymon configuration file

Using your favorite editor, append the following line into Snort.conf

(usually /etc/Snort/Snort.conf)

include Piggymon.conf

In case that you want to keep the Piggymon.conf file in another directory, please include the full path to it into the above line.

Step 5. Piggymon is ready to run!

The first time Piggymon is run, it will not display rate statistics, since it lacks the reference time pointer of the last run.

Before running Piggymon for the first time, remember to restart the Snort IDS, so that the new configuration options take effect.

(for debian systems)

sudo /etc/init.d/Snort restart

Try running Piggymon --help for available options. We strongly suggest the creation of a crontab entry, so that Piggymon is run automatically every 5 minutes.

Appendix B

Piggymon Rule Files

Stats_tracker.rules

 # Peter Politopoulos, 2011

 # This file contains the rules tracking flows not defined by well-known ports

 # Sources include protocol definitions, original p2p.rules file by Sourcefire (GPL), whitepapers and more.

 # original p2p.rules file, modified for snort_stat use

stats_tracker tcp any any -> any 8888 (msg:"Napster"; flow:established; content:"|00 02 00|"; depth:3; offset:1; flowbits:set,napster; flowbits:set,tracker_detected; sid:549; rev:8;)

stats_tracker tcp any any -> any 8888 (msg:"Napster"; flow:established; content:"|00 06 00|"; depth:3; offset:1; flowbits:set,napster; flowbits:set,tracker_detected; sid:550; rev:8;)

stats_tracker tcp any any -> any 8888 (msg:"Napster"; flow:established; content:"|00 CB 00|"; depth:3; offset:1; flowbits:set,napster; flowbits:set,tracker_detected; sid:551; rev:7;)

stats_tracker tcp any 8888 -> any any (msg:"Napster"; flow:established; content:"|00|_|02|"; depth:3; offset:1; flowbits:set,napster; flowbits:set,tracker_detected; sid:552; rev:7;)

stats_tracker tcp any any -> any any (msg:"GNUTella"; flow:established; content:"GNUTELLA"; depth:8; flowbits:set,gnutella; flowbits:set,tracker_detected; sid:1432; rev:6;)

stats_tracker tcp any any -> any any (msg:"GNUTella"; flow:established; content:"GNUTELLA CONNECT"; depth:40; flowbits:set,gnutella; flowbits:set,tracker_detected; sid:556; rev:5;)

stats_tracker tcp any any -> any any (msg:"GNUTella"; flow:established; content:"GNUTELLA OK"; depth:40; flowbits:set,gnutella; flowbits:set,tracker_detected; sid:557; rev:6;)

stats_tracker tcp any any <> any 6699 (msg:"Napster"; flow:established; content:".mp3"; nocase; flowbits:set,napster; flowbits:set,tracker_detected; sid:561; rev:6;)

stats_tracker tcp any any <> any 7777 (msg:"Napster"; flow:established; content:".mp3"; nocase; flowbits:set,napster; flowbits:set,tracker_detected; sid:562; rev:5;)

stats_tracker tcp any any <> any 6666 (msg:"Napster"; flow:established; content:".mp3"; nocase; flowbits:set,napster; flowbits:set,tracker_detected; sid:563; rev:6;)

stats_tracker tcp any any <> any 5555 (msg:"Napster"; flow:established; content:".mp3"; nocase; flowbits:set,napster; flowbits:set,tracker_detected; sid:564; rev:7;)

stats_tracker tcp any any <> any 8875 (msg:"Napster"; flow:established; content:"anon@napster.com"; flowbits:set,napster; flowbits:set,tracker_detected; sid:565; rev:6;)

stats_tracker tcp any any -> any 1214 (msg:"Fastrack kazaa-morpheus"; flow:established; content:"GET "; depth:4; reference:url,www.kazaa.com; reference:url,www.musiccity.com/technology.htm; flowbits:set,kazaa; flowbits:set,tracker_detected; sid:1383; rev:6;)

stats_tracker tcp any any -> any any (msg:"Fastrack kazaa-morpheus"; flow:established; content:"GET"; depth:3; content:"UserAgent|3A| KazaaClient"; reference:url,www.kazaa.com; flowbits:set,kazaa; flowbits:set,tracker_detected; sid: 1699; rev:7;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; flow:established; content:"GET"; depth:4; content:"/announce"; distance:1; content:"info_hash="; offset:4; content:"event=started"; offset:4; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:2180; rev:2;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; flow:established; content:"|13|BitTorrent protocol"; depth:20; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:2181; rev:2;)

stats_tracker tcp any any -> any 4242 (msg:"eDonkey"; flow:established; content:"|E3|"; depth:1; reference:url,www.kom.e-technik.tu-darmstadt.de/publications/abstracts/HB02-1.html; flowbits:set,edonkey; flowbits:set,tracker_detected; sid:2586; rev:2;)

stats_tracker tcp any 4711 -> any any (msg:"eDonkey"; flow:established; content:"Server|3A| eMule"; reference:url,www.emule-project.net; flowbits:set,edonkey; flowbits:set,tracker_detected; sid:2587; rev:2;)

stats_tracker udp any any -> any 41170 (msg:"Manolito"; content:"|01 02 00 14|"; depth:4; offset:16; reference:url,openlito.sourceforge.net; reference:url,www.blubster.com; flowbits:set,manolito; flowbits:set,tracker_detected; sid:3459; rev:3;)

 # BitTorrent

stats_tracker udp any any -> any any (msg:"BitTorrent"; content:"d1\:ad2\:id20\:"; depth:20; content:"ping"; distance:19; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909002; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"GET /scrape?info_hash="; depth:100; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909003; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"d1|3A|rd2|3A|id20|3A|"; depth:20; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909004; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"d1|3A|ad2|3A|id20|3A|"; depth:20; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909005; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_PIECE"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909006; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_REQUEST"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909007; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_CHOKE"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909008; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_UNCHOKE"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909009; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_HAVE"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909010; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_UNINTERESTED"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:190901; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_INTERESTER"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909012; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_BITFIELD"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909013; rev:1;)

stats_tracker tcp any any -> any any (msg:"BitTorrent"; content:"BT_CANCEL"; depth:50; flowbits:set,bittorrent; flowbits:set,tracker_detected; sid:1909014; rev:1;)

#Skype

stats_tracker tcp any any -> any any (msg:"Skype";flow:to_server,established; content:"|8046010301002d0000001000000500000400000a 0000090000640000620000080000030000060100800700c003 008006004002008004008 0|"; depth:112; flowbits:set,skype; flowbits:set,tracker_detected; sid:1909015; rev:1;)

#Does not track, before a login is actually initiated

stats_tracker tcp any any -> any any (msg:"Skype"; flow:to_server,established; dsize:5; content:"|16 03 01|"; depth:3; flowbits:set,skype_login; sid:1909016; rev:3;)

stats_tracker tcp any any -> any any (msg:"Skype"; flow:to_client,established; flowbits:isset,skype_login; dsize:5; content:"|17 03 01 00|"; depth:4; flowbits:set,skype; flowbits:set,tracker_detected; sid:1909017; rev:1;)

stats_tracker tcp any any -> any any (msg:"Skype"; flow:to_server,established; uricontent:"/ui/"; uricontent:"/getnewestversion"; content:"Host|3A| ui.skype.com"; flowbits:set,skype; flowbits:set,tracker_detected; sid:1909018; rev:5;)

stats_tracker tcp any 1024 -> any 33033 (msg:"Skype"; flow:to_server,established; content:"|17 03 01|"; depth:3; flowbits:set,skype; flowbits:set,tracker_detected; sid:1909019; rev:3;)

 # DC++

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|MyNick"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909020; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Lock EXTENDEDPROTOCOL"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909021; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Direction Download "; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909022; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Direction Upload "; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909023; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Supports"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909024; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|GetNickList|7C|"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909025; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|ValidateNick"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909026; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|ConnectToMe"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909027; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|HubName"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909028; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Hello"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909029; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|MyINFO $ALL"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909030; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|GetINFO"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909031; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Search Hub|3A|"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909032; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|OpList"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909033; rev:2;)

stats_tracker tcp any any -> any any (msg:"DirectConnect"; content:"|24|Sending"; depth:100; flowbits:set,dcpp; flowbits:set,tracker_detected; sid:1909034; rev:2;)

 # Edonkey

stats_tracker tcp any any -> any any (msg:"eDonkey"; content:"|030200707201000000|"; depth:100; flowbits:set,edonkey; flowbits:set,tracker_detected; sid:19090354; rev:1;)

 # Gnutella

stats_tracker tcp any any -> any any (msg:"GNUTella"; content:"GET /uri-res/N2R?urn\:sha1\:"; depth:100; flowbits:set,gnutella; flowbits:set,tracker_detected; sid:1909036; rev:2;)

stats_tracker tcp any any -> any any (msg:"GNUTella"; content:"Gnutella/"; depth:100; flowbits:set,gnutella; flowbits:set,tracker_detected; sid:1909037; rev:2;)

stats_tracker tcp any any -> any any (msg:"GNUTella"; content:"Server\: LimeWire/"; depth: 100; flowbits:set,gnutella; flowbits:set,tracker_detected; sid:1909038; rev:1;)

Stats_tracker_detected.rules

 # Peter Politopoulos, 2011

 # This file contains the rules matching flows already categorized by a tracker

stats_tracker_detected ip any any <> any any (msg:"BitTorrent"; sid:1100001; flowbits:isset,bittorrent; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"Napster"; sid:1100002; flowbits:isset,napster; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"GNUTella"; sid:1100003; flowbits:isset,gnutella; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"Kazaa"; sid:1100004; flowbits:isset,kazaa; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"eDonkey"; sid:1100005; flowbits:isset,edonkey; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"Manolito"; sid:1100006; flowbits:isset,manolito; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"skype"; sid:1100007; flowbits:isset,Skype; rev:1;)

stats_tracker_detected ip any any <> any any (msg:"DirectConnect"; sid:1100008; flowbits:isset,dcpp; rev:1;)

Stats_ports.rules

 # Peter Politopoulos, 2011

 # Ports 1-1023, mostly officially assigned

stats_ports tcp any any <> any 1 (msg:"TCPMUX"; flowbits:set,port_detected; sid:2000000; rev:1;)
stats_ports udp any any <> any 1 (msg:"TCPMUX"; flowbits:set,port_detected; sid:2000001; rev:1;)
stats_ports tcp any any <> any 2 (msg:"CompressNET"; flowbits:set,port_detected; sid:2000002; rev:1;)
stats_ports udp any any <> any 2 (msg:"CompressNET"; flowbits:set,port_detected; sid:2000003; rev:1;)
stats_ports tcp any any <> any 3 (msg:"CompressNET"; flowbits:set,port_detected; sid:2000004; rev:1;)
stats_ports udp any any <> any 3 (msg:"CompressNET"; flowbits:set,port_detected; sid:2000005; rev:1;)
stats_ports tcp any any <> any 5 (msg:"Remote Job Entry"; flowbits:set,port_detected; sid:2000006; rev:1;)
stats_ports udp any any <> any 5 (msg:"Remote Job Entry"; flowbits:set,port_detected; sid:2000007; rev:1;)
stats_ports tcp any any <> any 7 (msg:"Echo"; flowbits:set,port_detected; sid:2000008; rev:1;)
stats_ports udp any any <> any 7 (msg:"Echo"; flowbits:set,port_detected; sid:2000009; rev:1;)
stats_ports tcp any any <> any 9 (msg:"Discard"; flowbits:set,port_detected; sid:2000010; rev:1;)
stats_ports udp any any <> any 9 (msg:"Discard"; flowbits:set,port_detected; sid:2000011; rev:1;)
stats_ports tcp any any <> any 11 (msg:"systat"; flowbits:set,port_detected; sid:2000012; rev:1;)
stats_ports udp any any <> any 11 (msg:"systat"; flowbits:set,port_detected; sid:2000013; rev:1;)
stats_ports tcp any any <> any 13 (msg:"Daytime"; flowbits:set,port_detected; sid:2000014; rev:1;)
stats_ports udp any any <> any 13 (msg:"Daytime"; flowbits:set,port_detected; sid:2000015; rev:1;)
stats_ports tcp any any <> any 15 (msg:"netstat"; flowbits:set,port_detected; sid:2000016; rev:1;)
stats_ports udp any any <> any 15 (msg:"netstat"; flowbits:set,port_detected; sid:2000017; rev:1;)
stats_ports tcp any any <> any 17 (msg:"QotDay"; flowbits:set,port_detected; sid:2000018; rev:1;)
stats_ports udp any any <> any 17 (msg:"QotDay"; flowbits:set,port_detected; sid:2000019; rev:1;)
stats_ports tcp any any <> any 18 (msg:"Message Send Protocol"; flowbits:set,port_detected; sid:2000020; rev:1;)
stats_ports udp any any <> any 18 (msg:"Message Send Protocol"; flowbits:set,port_detected; sid:2000021; rev:1;)
stats_ports tcp any any <> any 19 (msg:"CHARGEN"; flowbits:set,port_detected; sid:2000022; rev:1;)
stats_ports udp any any <> any 19 (msg:"CHARGEN"; flowbits:set,port_detected; sid:2000023; rev:1;)
stats_ports tcp any any <> any 20 (msg:"FTP"; flowbits:set,port_detected; sid:2000024; rev:1;)
stats_ports tcp any any <> any 21 (msg:"FTP"; flowbits:set,port_detected; sid:2000025; rev:1;)
stats_ports tcp any any <> any 22 (msg:"SSH"; flowbits:set,port_detected; sid:2000026; rev:1;)
stats_ports udp any any <> any 22 (msg:"SSH"; flowbits:set,port_detected; sid:2000027; rev:1;)
stats_ports tcp any any <> any 23 (msg:"Telnet"; flowbits:set,port_detected; sid:2000028; rev:1;)
stats_ports tcp any any <> any 24 (msg:"Priv-mail"; flowbits:set,port_detected; sid:2000029; rev:1;)
stats_ports udp any any <> any 24 (msg:"Priv-mail"; flowbits:set,port_detected; sid:2000030; rev:1;)
stats_ports tcp any any <> any 25 (msg:"SMTP"; flowbits:set,port_detected; sid:2000031; rev:1;)
stats_ports tcp any any <> any 27 (msg:"NSW User System FE"; flowbits:set,port_detected; sid:2000032; rev:1;)
stats_ports udp any any <> any 27 (msg:"NSW User System FE"; flowbits:set,port_detected; sid:2000033; rev:1;)
stats_ports tcp any any <> any 34 (msg:"Remote File"; flowbits:set,port_detected; sid:2000034; rev:1;)
stats_ports udp any any <> any 34 (msg:"Remote File"; flowbits:set,port_detected; sid:2000035; rev:1;)
stats_ports tcp any any <> any 35 (msg:"Any private printer server protocol"; flowbits:set,port_detected; sid:2000036; rev:1;)
stats_ports udp any any <> any 35 (msg:"Any private printer server protocol"; flowbits:set,port_detected; sid:2000037; rev:1;)
stats_ports tcp any any <> any 37 (msg:"TIME"; flowbits:set,port_detected; sid:2000038; rev:1;)
stats_ports udp any any <> any 37 (msg:"TIME"; flowbits:set,port_detected; sid:2000039; rev:1;)
stats_ports tcp any any <> any 39 (msg:"RLP"; flowbits:set,port_detected; sid:2000040; rev:1;)
stats_ports udp any any <> any 39 (msg:"RLP"; flowbits:set,port_detected; sid:2000041; rev:1;)
stats_ports tcp any any <> any 41 (msg:"Graphics"; flowbits:set,port_detected; sid:2000042; rev:1;)
stats_ports udp any any <> any 41 (msg:"Graphics"; flowbits:set,port_detected; sid:2000043; rev:1;)
stats_ports tcp any any <> any 42 (msg:"WINS"; flowbits:set,port_detected; sid:2000044; rev:1;)
stats_ports udp any any <> any 42 (msg:"WINS"; flowbits:set,port_detected; sid:2000045; rev:1;)
stats_ports tcp any any <> any 43 (msg:"WHOIS protocol"; flowbits:set,port_detected; sid:2000046; rev:1;)
stats_ports tcp any any <> any 47 (msg:"NI FTP"; flowbits:set,port_detected; sid:2000047; rev:1;)
stats_ports udp any any <> any 47 (msg:"NI FTP"; flowbits:set,port_detected; sid:2000048; rev:1;)
stats_ports tcp any any <> any 49 (msg:"TACACS"; flowbits:set,port_detected; sid:2000049; rev:1;)
stats_ports udp any any <> any 49 (msg:"TACACS"; flowbits:set,port_detected; sid:2000050; rev:1;)

stats_ports tcp any any <> any 50 (msg:"RMCP"; flowbits:set,port_detected; sid:2000051; rev:1;)
stats_ports udp any any <> any 50 (msg:"RMCP"; flowbits:set,port_detected; sid:2000052; rev:1;)
stats_ports tcp any any <> any 51 (msg:"IMP Logical Address Maintenance"; flowbits:set,port_detected; sid:2000053; rev:1;)
stats_ports udp any any <> any 51 (msg:"IMP Logical Address Maintenance"; flowbits:set,port_detected; sid:2000054; rev:1;)
stats_ports tcp any any <> any 52 (msg:"XNS"; flowbits:set,port_detected; sid:2000055; rev:1;)
stats_ports udp any any <> any 52 (msg:"XNS"; flowbits:set,port_detected; sid:2000056; rev:1;)
stats_ports tcp any any <> any 53 (msg:"DNS"; flowbits:set,port_detected; sid:2000057; rev:1;)
stats_ports udp any any <> any 53 (msg:"DNS"; flowbits:set,port_detected; sid:2000058; rev:1;)
stats_ports tcp any any <> any 54 (msg:"XNS"; flowbits:set,port_detected; sid:2000059; rev:1;)
stats_ports udp any any <> any 54 (msg:"XNS"; flowbits:set,port_detected; sid:2000060; rev:1;)
stats_ports tcp any any <> any 55 (msg:"ISI-GL"; flowbits:set,port_detected; sid:2000061; rev:1;)
stats_ports udp any any <> any 55 (msg:"ISI-GL"; flowbits:set,port_detected; sid:2000062; rev:1;)
stats_ports tcp any any <> any 56 (msg:"RAP - XNS"; flowbits:set,port_detected; sid:2000063; rev:1;)
stats_ports udp any any <> any 56 (msg:"RAP - XNS"; flowbits:set,port_detected; sid:2000064; rev:1;)
stats_ports tcp any any <> any 57 (msg:"MTP"; flowbits:set,port_detected; sid:2000065; rev:1;)
stats_ports tcp any any <> any 58 (msg:"XNS"; flowbits:set,port_detected; sid:2000066; rev:1;)
stats_ports udp any any <> any 58 (msg:"XNS"; flowbits:set,port_detected; sid:2000067; rev:1;)
stats_ports udp any any <> any 67 (msg:"BOOTP - DHCP"; flowbits:set,port_detected; sid:2000068; rev:1;)
stats_ports udp any any <> any 68 (msg:"BOOTP - DHCP"; flowbits:set,port_detected; sid:2000069; rev:1;)
stats_ports udp any any <> any 69 (msg:"TFTP"; flowbits:set,port_detected; sid:2000070; rev:1;)
stats_ports tcp any any <> any 70 (msg:"Gopher"; flowbits:set,port_detected; sid:2000071; rev:1;)
stats_ports tcp any any <> any 71 (msg:"Genius"; flowbits:set,port_detected; sid:2000072; rev:1;)
stats_ports tcp any any <> any 79 (msg:"Finger"; flowbits:set,port_detected; sid:2000073; rev:1;)
stats_ports tcp any any <> any 80 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2000074; rev:1;)
stats_ports udp any any <> any 80 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2000075; rev:1;)
stats_ports tcp any any <> any 81 (msg:"Torpark"; flowbits:set,port_detected; sid:2000076; rev:1;)
stats_ports udp any any <> any 82 (msg:"Torpark"; flowbits:set,port_detected; sid:2000077; rev:1;)
stats_ports tcp any any <> any 83 (msg:"MIT ML Device"; flowbits:set,port_detected; sid:2000078; rev:1;)
stats_ports tcp any any <> any 88 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000079; rev:1;)
stats_ports udp any any <> any 88 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000080; rev:1;)
stats_ports tcp any any <> any 90 (msg:"dnsix"; flowbits:set,port_detected; sid:2000081; rev:1;)
stats_ports udp any any <> any 90 (msg:"dnsix"; flowbits:set,port_detected; sid:2000082; rev:1;)
stats_ports tcp any any <> any 99 (msg:"WIP"; flowbits:set,port_detected; sid:2000083; rev:1;)
stats_ports tcp any any <> any 101 (msg:"NIC host name"; flowbits:set,port_detected; sid:2000084; rev:1;)
stats_ports tcp any any <> any 102 (msg:"ISO-TSAP"; flowbits:set,port_detected; sid:2000085; rev:1;)
stats_ports tcp any any <> any 104 (msg:"ACR - NEM"; flowbits:set,port_detected; sid:2000086; rev:1;)
stats_ports udp any any <> any 104 (msg:"ACR - NEM"; flowbits:set,port_detected; sid:2000087; rev:1;)
stats_ports tcp any any <> any 105 (msg:"CCSO"; flowbits:set,port_detected; sid:2000088; rev:1;)
stats_ports udp any any <> any 105 (msg:"CCSO"; flowbits:set,port_detected; sid:2000089; rev:1;)
stats_ports tcp any any <> any 107 (msg:"TELNET"; flowbits:set,port_detected; sid:2000090; rev:1;)
stats_ports tcp any any <> any 108 (msg:"SNA"; flowbits:set,port_detected; sid:2000091; rev:1;)
stats_ports udp any any <> any 108 (msg:"SNA"; flowbits:set,port_detected; sid:2000092; rev:1;)
stats_ports tcp any any <> any 109 (msg:"POP"; flowbits:set,port_detected; sid:2000093; rev:1;)
stats_ports tcp any any <> any 110 (msg:"POP"; flowbits:set,port_detected; sid:2000094; rev:1;)
stats_ports tcp any any <> any 111 (msg:"SunRPC"; flowbits:set,port_detected; sid:2000095; rev:1;)
stats_ports udp any any <> any 111 (msg:"SunRPC"; flowbits:set,port_detected; sid:2000096; rev:1;)
stats_ports udp any any <> any 113 (msg:"auth"; flowbits:set,port_detected; sid:2000097; rev:1;)
stats_ports tcp any any <> any 113 (msg:"ident"; flowbits:set,port_detected; sid:2000098; rev:1;)
stats_ports tcp any any <> any 115 (msg:"SFTP"; flowbits:set,port_detected; sid:2000099; rev:1;)
stats_ports tcp any any <> any 117 (msg:"UUCP"; flowbits:set,port_detected; sid:2000100; rev:1;)
stats_ports tcp any any <> any 118 (msg:"SQL"; flowbits:set,port_detected; sid:2000101; rev:1;)
stats_ports udp any any <> any 118 (msg:"SQL"; flowbits:set,port_detected; sid:2000102; rev:1;)

stats_ports tcp any any <> any 119 (msg:"NNTP"; flowbits:set,port_detected; sid:2000103; rev:1;)
stats_ports udp any any <> any 123 (msg:"NTP"; flowbits:set,port_detected; sid:2000104; rev:1;)
stats_ports tcp any any <> any 135 (msg:"DCE-RPC"; flowbits:set,port_detected; sid:2000105; rev:1;)
stats_ports udp any any <> any 135 (msg:"DCE-RPC"; flowbits:set,port_detected; sid:2000106; rev:1;)
stats_ports tcp any any <> any 137 (msg:"NetBIOS"; flowbits:set,port_detected; sid:2000107; rev:1;)
stats_ports udp any any <> any 137 (msg:"NetBIOS"; flowbits:set,port_detected; sid:2000108; rev:1;)
stats_ports tcp any any <> any 138 (msg:"NetBIOS"; flowbits:set,port_detected; sid:2000109; rev:1;)
stats_ports udp any any <> any 138 (msg:"NetBIOS"; flowbits:set,port_detected; sid:2000110; rev:1;)
stats_ports tcp any any <> any 139 (msg:"NetBIOS"; flowbits:set,port_detected; sid:2000111; rev:1;)
stats_ports udp any any <> any 139 (msg:"NetBIOS"; flowbits:set,port_detected; sid:2000112; rev:1;)
stats_ports tcp any any <> any 143 (msg:"IMAP"; flowbits:set,port_detected; sid:2000113; rev:1;)
stats_ports udp any any <> any 143 (msg:"IMAP"; flowbits:set,port_detected; sid:2000114; rev:1;)
stats_ports tcp any any <> any 152 (msg:"BFTP"; flowbits:set,port_detected; sid:2000115; rev:1;)
stats_ports udp any any <> any 152 (msg:"BFTP"; flowbits:set,port_detected; sid:2000116; rev:1;)
stats_ports tcp any any <> any 153 (msg:"SGMP"; flowbits:set,port_detected; sid:2000117; rev:1;)
stats_ports udp any any <> any 153 (msg:"SGMP"; flowbits:set,port_detected; sid:2000118; rev:1;)
stats_ports tcp any any <> any 156 (msg:"SQL"; flowbits:set,port_detected; sid:2000119; rev:1;)
stats_ports udp any any <> any 156 (msg:"SQL"; flowbits:set,port_detected; sid:2000120; rev:1;)
stats_ports tcp any any <> any 158 (msg:"DMSP"; flowbits:set,port_detected; sid:2000121; rev:1;)
stats_ports udp any any <> any 158 (msg:"DMSP"; flowbits:set,port_detected; sid:2000122; rev:1;)
stats_ports udp any any <> any 161 (msg:"SNMP"; flowbits:set,port_detected; sid:2000123; rev:1;)
stats_ports tcp any any <> any 162 (msg:"SNMP"; flowbits:set,port_detected; sid:2000124; rev:1;)
stats_ports udp any any <> any 162 (msg:"SNMP"; flowbits:set,port_detected; sid:2000125; rev:1;)
stats_ports tcp any any <> any 170 (msg:"Print-srv"; flowbits:set,port_detected; sid:2000126; rev:1;)
stats_ports tcp any any <> any 177 (msg:"XDMCP"; flowbits:set,port_detected; sid:2000127; rev:1;)
stats_ports udp any any <> any 177 (msg:"XDMCP"; flowbits:set,port_detected; sid:2000128; rev:1;)
stats_ports tcp any any <> any 179 (msg:"BGP"; flowbits:set,port_detected; sid:2000129; rev:1;)
stats_ports tcp any any <> any 194 (msg:"IRC"; flowbits:set,port_detected; sid:2000130; rev:1;)
stats_ports udp any any <> any 194 (msg:"IRC"; flowbits:set,port_detected; sid:2000131; rev:1;)
stats_ports tcp any any <> any 199 (msg:"SMUX"; flowbits:set,port_detected; sid:2000132; rev:1;)
stats_ports udp any any <> any 199 (msg:"SMUX"; flowbits:set,port_detected; sid:2000133; rev:1;)
stats_ports tcp any any <> any 201 (msg:"AppleTalk Routing Maintenance"; flowbits:set,port_detected; sid:2000134; rev:1;)
stats_ports udp any any <> any 201 (msg:"AppleTalk Routing Maintenance"; flowbits:set,port_detected; sid:2000135; rev:1;)
stats_ports tcp any any <> any 209 (msg:"Quick Mail"; flowbits:set,port_detected; sid:2000136; rev:1;)
stats_ports udp any any <> any 209 (msg:"Quick Mail"; flowbits:set,port_detected; sid:2000137; rev:1;)
stats_ports tcp any any <> any 210 (msg:"ANSI Z39.50"; flowbits:set,port_detected; sid:2000138; rev:1;)
stats_ports udp any any <> any 210 (msg:"ANSI Z39.50"; flowbits:set,port_detected; sid:2000139; rev:1;)
stats_ports tcp any any <> any 213 (msg:"IPX"; flowbits:set,port_detected; sid:2000140; rev:1;)
stats_ports udp any any <> any 213 (msg:"IPX"; flowbits:set,port_detected; sid:2000141; rev:1;)
stats_ports tcp any any <> any 218 (msg:"MPP"; flowbits:set,port_detected; sid:2000142; rev:1;)
stats_ports udp any any <> any 218 (msg:"MPP"; flowbits:set,port_detected; sid:2000143; rev:1;)
stats_ports tcp any any <> any 220 (msg:"IMAP"; flowbits:set,port_detected; sid:2000144; rev:1;)
stats_ports udp any any <> any 220 (msg:"IMAP"; flowbits:set,port_detected; sid:2000145; rev:1;)
stats_ports tcp any any <> any 256 (msg:"2DEV 2SP"; flowbits:set,port_detected; sid:2000146; rev:1;)
stats_ports udp any any <> any 256 (msg:"2DEV 2SP"; flowbits:set,port_detected; sid:2000147; rev:1;)
stats_ports tcp any any <> any 259 (msg:"ESRO"; flowbits:set,port_detected; sid:2000148; rev:1;)
stats_ports udp any any <> any 259 (msg:"ESRO"; flowbits:set,port_detected; sid:2000149; rev:1;)
stats_ports tcp any any <> any 264 (msg:"BGMP"; flowbits:set,port_detected; sid:2000150; rev:1;)
stats_ports udp any any <> any 264 (msg:"BGMP"; flowbits:set,port_detected; sid:2000151; rev:1;)
stats_ports tcp any any <> any 280 (msg:"http-mgmt"; flowbits:set,port_detected; sid:2000152; rev:1;)
stats_ports udp any any <> any 280 (msg:"http-mgmt"; flowbits:set,port_detected; sid:2000153; rev:1;)
stats_ports tcp any any <> any 308 (msg:"Novastor Online Backup"; flowbits:set,port_detected; sid:2000154; rev:1;)

stats_ports tcp any any <> any 311 (msg:"MacOSX Srvr Adm"; flowbits:set,port_detected; sid:2000155; rev:1;)
stats_ports tcp any any <> any 318 (msg:"Time Stamp"; flowbits:set,port_detected; sid:2000156; rev:1;)
stats_ports udp any any <> any 318 (msg:"Time Stamp"; flowbits:set,port_detected; sid:2000157; rev:1;)
stats_ports udp any any <> any 319 (msg:"Precision time"; flowbits:set,port_detected; sid:2000158; rev:1;)
stats_ports udp any any <> any 320 (msg:"Precision time"; flowbits:set,port_detected; sid:2000159; rev:1;)
stats_ports tcp any any <> any 323 (msg:"IMMP"; flowbits:set,port_detected; sid:2000160; rev:1;)
stats_ports udp any any <> any 323 (msg:"IMMP"; flowbits:set,port_detected; sid:2000161; rev:1;)
stats_ports tcp any any <> any 350 (msg:"MATIP"; flowbits:set,port_detected; sid:2000162; rev:1;)
stats_ports udp any any <> any 350 (msg:"MATIP"; flowbits:set,port_detected; sid:2000163; rev:1;)
stats_ports tcp any any <> any 351 (msg:"MATIP"; flowbits:set,port_detected; sid:2000164; rev:1;)
stats_ports udp any any <> any 351 (msg:"MATIP"; flowbits:set,port_detected; sid:2000165; rev:1;)
stats_ports tcp any any <> any 366 (msg:"ODMR"; flowbits:set,port_detected; sid:2000166; rev:1;)
stats_ports udp any any <> any 366 (msg:"ODMR"; flowbits:set,port_detected; sid:2000167; rev:1;)
stats_ports tcp any any <> any 369 (msg:"Rpc2portmap"; flowbits:set,port_detected; sid:2000168; rev:1;)
stats_ports udp any any <> any 369 (msg:"Rpc2portmap"; flowbits:set,port_detected; sid:2000169; rev:1;)
stats_ports tcp any any <> any 370 (msg:"codaauth2 - securecast"; flowbits:set,port_detected; sid:2000170; rev:1;)
stats_ports udp any any <> any 370 (msg:"codaauth2 - securecast"; flowbits:set,port_detected; sid:2000171; rev:1;)
stats_ports tcp any any <> any 371 (msg:"ClearCase albd"; flowbits:set,port_detected; sid:2000172; rev:1;)
stats_ports udp any any <> any 371 (msg:"ClearCase albd"; flowbits:set,port_detected; sid:2000173; rev:1;)
stats_ports tcp any any <> any 383 (msg:"HP data alarm manager"; flowbits:set,port_detected; sid:2000174; rev:1;)
stats_ports udp any any <> any 383 (msg:"HP data alarm manager"; flowbits:set,port_detected; sid:2000175; rev:1;)
stats_ports tcp any any <> any 384 (msg:"A Remote Network Server System"; flowbits:set,port_detected; sid:2000176; rev:1;)
stats_ports udp any any <> any 384 (msg:"A Remote Network Server System"; flowbits:set,port_detected; sid:2000177; rev:1;)
stats_ports tcp any any <> any 387 (msg:"AURP"; flowbits:set,port_detected; sid:2000178; rev:1;)
stats_ports udp any any <> any 387 (msg:"AURP"; flowbits:set,port_detected; sid:2000179; rev:1;)
stats_ports tcp any any <> any 389 (msg:"LDAP"; flowbits:set,port_detected; sid:2000180; rev:1;)
stats_ports udp any any <> any 389 (msg:"LDAP"; flowbits:set,port_detected; sid:2000181; rev:1;)
stats_ports tcp any any <> any 401 (msg:"UPS"; flowbits:set,port_detected; sid:2000182; rev:1;)
stats_ports udp any any <> any 401 (msg:"UPS"; flowbits:set,port_detected; sid:2000183; rev:1;)
stats_ports tcp any any <> any 402 (msg:"Altiris, Altiris Deployment Client"; flowbits:set,port_detected; sid:2000184; rev:1;)
stats_ports tcp any any <> any 411 (msg:"DC++ - DC"; flowbits:set,port_detected; sid:2000185; rev:1;)
stats_ports tcp any any <> any 412 (msg:"DC++ - DC"; flowbits:set,port_detected; sid:2000186; rev:1;)
stats_ports tcp any any <> any 427 (msg:"SLP"; flowbits:set,port_detected; sid:2000187; rev:1;)
stats_ports udp any any <> any 427 (msg:"SLP"; flowbits:set,port_detected; sid:2000188; rev:1;)
stats_ports tcp any any <> any 443 (msg:"HTTPS"; flowbits:set,port_detected; sid:2000189; rev:1;)
stats_ports tcp any any <> any 444 (msg:"SNPP"; flowbits:set,port_detected; sid:2000190; rev:1;)
stats_ports udp any any <> any 444 (msg:"SNPP"; flowbits:set,port_detected; sid:2000191; rev:1;)
stats_ports tcp any any <> any 445 (msg:"Microsoft-DS"; flowbits:set,port_detected; sid:2000192; rev:1;)
stats_ports tcp any any <> any 464 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000193; rev:1;)
stats_ports udp any any <> any 464 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000194; rev:1;)
stats_ports tcp any any <> any 465 (msg:"SMTP"; flowbits:set,port_detected; sid:2000195; rev:1;)
stats_ports tcp any any <> any 475 (msg:"tcpnethaspsrv"; flowbits:set,port_detected; sid:2000196; rev:1;)
stats_ports udp any any <> any 475 (msg:"tcpnethaspsrv"; flowbits:set,port_detected; sid:2000197; rev:1;)
stats_ports tcp any any <> any 497 (msg:"Dantz Retrospect"; flowbits:set,port_detected; sid:2000198; rev:1;)
stats_ports udp any any <> any 500 (msg:"ISAKMP"; flowbits:set,port_detected; sid:2000199; rev:1;)
stats_ports tcp any any <> any 501 (msg:"STMF"; flowbits:set,port_detected; sid:2000200; rev:1;)
stats_ports tcp any any <> any 502 (msg:"asa-appl-proto - Modbus"; flowbits:set,port_detected; sid:2000201; rev:1;)
stats_ports udp any any <> any 502 (msg:"asa-appl-proto - Modbus"; flowbits:set,port_detected; sid:2000202; rev:1;)
stats_ports tcp any any <> any 504 (msg:"Citadel"; flowbits:set,port_detected; sid:2000203; rev:1;)
stats_ports udp any any <> any 504 (msg:"Citadel"; flowbits:set,port_detected; sid:2000204; rev:1;)
stats_ports tcp any any <> any 510 (msg:"First Class"; flowbits:set,port_detected; sid:2000205; rev:1;)
stats_ports udp any any <> any 512 (msg:"comsat - biff"; flowbits:set,port_detected; sid:2000206; rev:1;)

stats_ports tcp any any <> any 512 (msg:"Rexec"; flowbits:set,port_detected; sid:2000207; rev:1;)
stats_ports tcp any any <> any 513 (msg:"rlogin"; flowbits:set,port_detected; sid:2000208; rev:1;)
stats_ports udp any any <> any 513 (msg:"Who"; flowbits:set,port_detected; sid:2000209; rev:1;)
stats_ports tcp any any <> any 514 (msg:"rsh"; flowbits:set,port_detected; sid:2000210; rev:1;)
stats_ports udp any any <> any 514 (msg:"Syslog"; flowbits:set,port_detected; sid:2000211; rev:1;)
stats_ports tcp any any <> any 515 (msg:"LPD"; flowbits:set,port_detected; sid:2000212; rev:1;)
stats_ports udp any any <> any 517 (msg:"Talk"; flowbits:set,port_detected; sid:2000213; rev:1;)
stats_ports udp any any <> any 518 (msg:"NTalk"; flowbits:set,port_detected; sid:2000214; rev:1;)
stats_ports tcp any any <> any 520 (msg:"efs"; flowbits:set,port_detected; sid:2000215; rev:1;)
stats_ports udp any any <> any 520 (msg:"RIP"; flowbits:set,port_detected; sid:2000216; rev:1;)
stats_ports tcp any any <> any 524 (msg:"NCP"; flowbits:set,port_detected; sid:2000217; rev:1;)
stats_ports udp any any <> any 524 (msg:"NCP"; flowbits:set,port_detected; sid:2000218; rev:1;)
stats_ports udp any any <> any 525 (msg:"Timed"; flowbits:set,port_detected; sid:2000219; rev:1;)
stats_ports tcp any any <> any 530 (msg:"RPC"; flowbits:set,port_detected; sid:2000220; rev:1;)
stats_ports udp any any <> any 530 (msg:"RPC"; flowbits:set,port_detected; sid:2000221; rev:1;)
stats_ports tcp any any <> any 531 (msg:"AOL IM"; flowbits:set,port_detected; sid:2000222; rev:1;)
stats_ports udp any any <> any 531 (msg:"AOL IM"; flowbits:set,port_detected; sid:2000223; rev:1;)
stats_ports tcp any any <> any 532 (msg:"netnews"; flowbits:set,port_detected; sid:2000224; rev:1;)
stats_ports udp any any <> any 533 (msg:"netwall"; flowbits:set,port_detected; sid:2000225; rev:1;)
stats_ports tcp any any <> any 540 (msg:"UUCP"; flowbits:set,port_detected; sid:2000226; rev:1;)
stats_ports tcp any any <> any 542 (msg:"commerce"; flowbits:set,port_detected; sid:2000227; rev:1;)
stats_ports udp any any <> any 542 (msg:"commerce"; flowbits:set,port_detected; sid:2000228; rev:1;)
stats_ports tcp any any <> any 543 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000229; rev:1;)
stats_ports tcp any any <> any 544 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000230; rev:1;)
stats_ports tcp any any <> any 545 (msg:"OSIsoft PI"; flowbits:set,port_detected; sid:2000231; rev:1;)
stats_ports tcp any any <> any 546 (msg:"DHCP"; flowbits:set,port_detected; sid:2000232; rev:1;)
stats_ports udp any any <> any 546 (msg:"DHCP"; flowbits:set,port_detected; sid:2000233; rev:1;)
stats_ports tcp any any <> any 547 (msg:"DHCP"; flowbits:set,port_detected; sid:2000234; rev:1;)
stats_ports udp any any <> any 547 (msg:"DHCP"; flowbits:set,port_detected; sid:2000235; rev:1;)
stats_ports tcp any any <> any 548 (msg:"AFP"; flowbits:set,port_detected; sid:2000236; rev:1;)
stats_ports udp any any <> any 550 (msg:"new-rwho, new-who"; flowbits:set,port_detected; sid:2000237; rev:1;)
stats_ports tcp any any <> any 554 (msg:"RTSP"; flowbits:set,port_detected; sid:2000238; rev:1;)
stats_ports udp any any <> any 554 (msg:"RTSP"; flowbits:set,port_detected; sid:2000239; rev:1;)
stats_ports tcp any any <> any 556 (msg:"RFS"; flowbits:set,port_detected; sid:2000240; rev:1;)
stats_ports udp any any <> any 560 (msg:"rmonitor"; flowbits:set,port_detected; sid:2000241; rev:1;)
stats_ports udp any any <> any 561 (msg:"monitor"; flowbits:set,port_detected; sid:2000242; rev:1;)
stats_ports tcp any any <> any 563 (msg:"NNTPS"; flowbits:set,port_detected; sid:2000243; rev:1;)
stats_ports udp any any <> any 563 (msg:"NNTPS"; flowbits:set,port_detected; sid:2000244; rev:1;)
stats_ports tcp any any <> any 587 (msg:"SMTP"; flowbits:set,port_detected; sid:2000245; rev:1;)
stats_ports tcp any any <> any 591 (msg:"FileMaker"; flowbits:set,port_detected; sid:2000246; rev:1;)
stats_ports tcp any any <> any 593 (msg:"HTTP RPC - MS Exchange"; flowbits:set,port_detected; sid:2000247; rev:1;)
stats_ports udp any any <> any 593 (msg:"HTTP RPC - MS Exchange"; flowbits:set,port_detected; sid:2000248; rev:1;)
stats_ports tcp any any <> any 604 (msg:"TUNNEL"; flowbits:set,port_detected; sid:2000249; rev:1;)
stats_ports udp any any <> any 623 (msg:"ASF-RMCP"; flowbits:set,port_detected; sid:2000250; rev:1;)
stats_ports tcp any any <> any 631 (msg:"CUPS - IPP"; flowbits:set,port_detected; sid:2000251; rev:1;)
stats_ports udp any any <> any 631 (msg:"CUPS - IPP"; flowbits:set,port_detected; sid:2000252; rev:1;)
stats_ports tcp any any <> any 635 (msg:"RLZ DBase"; flowbits:set,port_detected; sid:2000253; rev:1;)
stats_ports udp any any <> any 635 (msg:"RLZ DBase"; flowbits:set,port_detected; sid:2000254; rev:1;)
stats_ports tcp any any <> any 636 (msg:"LDAPS"; flowbits:set,port_detected; sid:2000255; rev:1;)
stats_ports udp any any <> any 636 (msg:"LDAPS"; flowbits:set,port_detected; sid:2000256; rev:1;)
stats_ports tcp any any <> any 639 (msg:"MSDP"; flowbits:set,port_detected; sid:2000257; rev:1;)
stats_ports udp any any <> any 639 (msg:"MSDP"; flowbits:set,port_detected; sid:2000258; rev:1;)

stats_ports tcp any any <> any 641 (msg:"Nexus Remote Command"; flowbits:set,port_detected; sid:2000259; rev:1;)

stats_ports udp any any <> any 641 (msg:"Nexus Remote Command"; flowbits:set,port_detected; sid:2000260; rev:1;)

stats_ports tcp any any <> any 646 (msg:"LDP"; flowbits:set,port_detected; sid:2000261; rev:1;)

stats_ports udp any any <> any 646 (msg:"LDP"; flowbits:set,port_detected; sid:2000262; rev:1;)

stats_ports tcp any any <> any 647 (msg:"DHCP"; flowbits:set,port_detected; sid:2000263; rev:1;)

stats_ports tcp any any <> any 648 (msg:"RRP"; flowbits:set,port_detected; sid:2000264; rev:1;)

stats_ports tcp any any <> any 651 (msg:"IEEE-MMS"; flowbits:set,port_detected; sid:2000265; rev:1;)

stats_ports udp any any <> any 651 (msg:"IEEE-MMS"; flowbits:set,port_detected; sid:2000266; rev:1;)

stats_ports tcp any any <> any 652 (msg:"DTCP"; flowbits:set,port_detected; sid:2000267; rev:1;)

stats_ports tcp any any <> any 653 (msg:"Nexus Remote Command"; flowbits:set,port_detected; sid:2000268; rev:1;)

stats_ports udp any any <> any 653 (msg:"Nexus Remote Command"; flowbits:set,port_detected; sid:2000269; rev:1;)

stats_ports tcp any any <> any 654 (msg:"MMS - MMP"; flowbits:set,port_detected; sid:2000270; rev:1;)

stats_ports tcp any any <> any 657 (msg:"IBM RMC"; flowbits:set,port_detected; sid:2000271; rev:1;)

stats_ports udp any any <> any 657 (msg:"IBM RMC"; flowbits:set,port_detected; sid:2000272; rev:1;)

stats_ports tcp any any <> any 660 (msg:"Mac OS X Server Admin"; flowbits:set,port_detected; sid:2000273; rev:1;)

stats_ports tcp any any <> any 665 (msg:"sun-dr"; flowbits:set,port_detected; sid:2000274; rev:1;)

stats_ports udp any any <> any 666 (msg:"Doom"; flowbits:set,port_detected; sid:2000275; rev:1;)

stats_ports tcp any any <> any 674 (msg:"ACAP"; flowbits:set,port_detected; sid:2000276; rev:1;)

stats_ports tcp any any <> any 691 (msg:"MS Exchange Routing"; flowbits:set,port_detected; sid:2000277; rev:1;)

stats_ports tcp any any <> any 692 (msg:"Hyperwave-ISP"; flowbits:set,port_detected; sid:2000278; rev:1;)

stats_ports tcp any any <> any 694 (msg:"Linux-HA"; flowbits:set,port_detected; sid:2000279; rev:1;)

stats_ports udp any any <> any 694 (msg:"Linux-HA"; flowbits:set,port_detected; sid:2000280; rev:1;)

stats_ports tcp any any <> any 695 (msg:"IEEE-MMS-SSL"; flowbits:set,port_detected; sid:2000281; rev:1;)

stats_ports udp any any <> any 698 (msg:"OLSR"; flowbits:set,port_detected; sid:2000282; rev:1;)

stats_ports tcp any any <> any 699 (msg:"Access Network"; flowbits:set,port_detected; sid:2000283; rev:1;)

stats_ports tcp any any <> any 700 (msg:"EPP"; flowbits:set,port_detected; sid:2000284; rev:1;)

stats_ports tcp any any <> any 701 (msg:"LMP"; flowbits:set,port_detected; sid:2000285; rev:1;)

stats_ports tcp any any <> any 702 (msg:"IRIS"; flowbits:set,port_detected; sid:2000286; rev:1;)

stats_ports tcp any any <> any 706 (msg:"SILC"; flowbits:set,port_detected; sid:2000287; rev:1;)

stats_ports tcp any any <> any 711 (msg:"Cisco Tag Distribution"; flowbits:set,port_detected; sid:2000288; rev:1;)

stats_ports udp any any <> any 712 (msg:"Promise RAID Controller"; flowbits:set,port_detected; sid:2000289; rev:1;)

stats_ports tcp any any <> any 712 (msg:"TBRPF"; flowbits:set,port_detected; sid:2000290; rev:1;)

stats_ports tcp any any <> any 720 (msg:"SMQP"; flowbits:set,port_detected; sid:2000291; rev:1;)

stats_ports tcp any any <> any 749 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000292; rev:1;)

stats_ports udp any any <> any 749 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000293; rev:1;)

stats_ports udp any any <> any 750 (msg:"Kerberos - loadav"; flowbits:set,port_detected; sid:2000294; rev:1;)

stats_ports tcp any any <> any 750 (msg:"rfile"; flowbits:set,port_detected; sid:2000295; rev:1;)

stats_ports tcp any any <> any 751 (msg:"Kerberos - pump"; flowbits:set,port_detected; sid:2000296; rev:1;)

stats_ports udp any any <> any 751 (msg:"Kerberos - pump"; flowbits:set,port_detected; sid:2000297; rev:1;)

stats_ports udp any any <> any 752 (msg:"passwd_server"; flowbits:set,port_detected; sid:2000298; rev:1;)

stats_ports tcp any any <> any 752 (msg:"qrh"; flowbits:set,port_detected; sid:2000299; rev:1;)

stats_ports tcp any any <> any 753 (msg:"rrh"; flowbits:set,port_detected; sid:2000300; rev:1;)

stats_ports udp any any <> any 753 (msg:"rrh"; flowbits:set,port_detected; sid:2000301; rev:1;)

stats_ports tcp any any <> any 754 (msg:"Kerberos"; flowbits:set,port_detected; sid:2000302; rev:1;)

stats_ports udp any any <> any 754 (msg:"tell send"; flowbits:set,port_detected; sid:2000303; rev:1;)

stats_ports tcp any any <> any 760 (msg:"ns"; flowbits:set,port_detected; sid:2000304; rev:1;)

stats_ports udp any any <> any 760 (msg:"ns"; flowbits:set,port_detected; sid:2000305; rev:1;)

stats_ports tcp any any <> any 782 (msg:"Conserver serial-console management server"; flowbits:set,port_detected; sid: 2000306; rev:1;)

stats_ports tcp any any <> any 783 (msg:"spamd"; flowbits:set,port_detected; sid:2000307; rev:1;)

stats_ports tcp any any <> any 829 (msg:"CMP"; flowbits:set,port_detected; sid:2000308; rev:1;)

stats_ports tcp any any <> any 843 (msg:"Adobe Flash socket policy server"; flowbits:set,port_detected; sid:2000309; rev:1;)

stats_ports tcp any any <> any 847 (msg:"DHCP"; flowbits:set,port_detected; sid:2000310; rev:1;)
stats_ports tcp any any <> any 860 (msg:"iSCSI"; flowbits:set,port_detected; sid:2000311; rev:1;)
stats_ports tcp any any <> any 873 (msg:"rsync"; flowbits:set,port_detected; sid:2000312; rev:1;)
stats_ports tcp any any <> any 888 (msg:"cddbp"; flowbits:set,port_detected; sid:2000313; rev:1;)
stats_ports tcp any any <> any 901 (msg:"VMware - SWAT"; flowbits:set,port_detected; sid:2000314; rev:1;)
stats_ports udp any any <> any 901 (msg:"VMware - SWAT"; flowbits:set,port_detected; sid:2000315; rev:1;)
stats_ports udp any any <> any 902 (msg:"ideafarm-door"; flowbits:set,port_detected; sid:2000316; rev:1;)
stats_ports tcp any any <> any 902 (msg:"ideafarm-door - VMware"; flowbits:set,port_detected; sid:2000317; rev:1;)
stats_ports udp any any <> any 902 (msg:"ideafarm-door - VMware"; flowbits:set,port_detected; sid:2000318; rev:1;)
stats_ports tcp any any <> any 903 (msg:"VMware"; flowbits:set,port_detected; sid:2000319; rev:1;)
stats_ports tcp any any <> any 904 (msg:"VMware"; flowbits:set,port_detected; sid:2000320; rev:1;)
stats_ports tcp any any <> any 911 (msg:"NCA"; flowbits:set,port_detected; sid:2000321; rev:1;)
stats_ports tcp any any <> any 953 (msg:"RNDC"; flowbits:set,port_detected; sid:2000322; rev:1;)
stats_ports udp any any <> any 953 (msg:"RNDC"; flowbits:set,port_detected; sid:2000323; rev:1;)
stats_ports tcp any any <> any 981 (msg:"Check Point fw"; flowbits:set,port_detected; sid:2000324; rev:1;)
stats_ports tcp any any <> any 987 (msg:"MS Sharepoint"; flowbits:set,port_detected; sid:2000325; rev:1;)
stats_ports tcp any any <> any 989 (msg:"FTPS"; flowbits:set,port_detected; sid:2000326; rev:1;)
stats_ports udp any any <> any 989 (msg:"FTPS"; flowbits:set,port_detected; sid:2000327; rev:1;)
stats_ports tcp any any <> any 990 (msg:"FTPS"; flowbits:set,port_detected; sid:2000328; rev:1;)
stats_ports udp any any <> any 990 (msg:"FTPS"; flowbits:set,port_detected; sid:2000329; rev:1;)
stats_ports tcp any any <> any 991 (msg:"NAS"; flowbits:set,port_detected; sid:2000330; rev:1;)
stats_ports udp any any <> any 991 (msg:"NAS"; flowbits:set,port_detected; sid:2000331; rev:1;)
stats_ports tcp any any <> any 992 (msg:"TELNET over SSL"; flowbits:set,port_detected; sid:2000332; rev:1;)
stats_ports udp any any <> any 992 (msg:"TELNET over SSL"; flowbits:set,port_detected; sid:2000333; rev:1;)
stats_ports tcp any any <> any 993 (msg:"IMAPS"; flowbits:set,port_detected; sid:2000334; rev:1;)
stats_ports tcp any any <> any 995 (msg:"POP"; flowbits:set,port_detected; sid:2000335; rev:1;)
stats_ports tcp any any <> any 999 (msg:"ScimoreDB"; flowbits:set,port_detected; sid:2000336; rev:1;)
stats_ports tcp any any <> any 1001 (msg:"JtoMB"; flowbits:set,port_detected; sid:2000337; rev:1;)
stats_ports tcp any any <> any 1002 (msg:"cogbot"; flowbits:set,port_detected; sid:2000338; rev:1;)

Stats_ports_multi.rules

 # Peter Politopoulos, 2011
 # Port ranges and multi-port applications

stats_ports tcp any any <> any 32459 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2000343; rev:1;)
stats_ports udp any any <> any 32459 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2000344; rev:1;)
stats_ports tcp any any <> any 4662 (msg:"eDonkey"; flowbits:set,port_detected; sid:2000345; rev:1;)
stats_ports udp any any <> any 4662 (msg:"eDonkey"; flowbits:set,port_detected; sid:2000346; rev:1;)
stats_ports tcp any any <> any 6660:6669 (msg:"IRC"; flowbits:set,port_detected; sid:2000347; rev:1;)
stats_ports udp any any <> any 6660:6669 (msg:"IRC"; flowbits:set,port_detected; sid:2000348; rev:1;)
stats_ports tcp any any <> any 1119 (msg:"WoW"; flowbits:set,port_detected; sid:2000351; rev:1;)
stats_ports tcp any any <> any 3724 (msg:"WoW"; flowbits:set,port_detected; sid:2000352; rev:1;)
stats_ports tcp any any <> any 6112:6114 (msg:"WoW"; flowbits:set,port_detected; sid:2000353; rev:1;)
stats_ports tcp any any <> any 4000 (msg:"WoW"; flowbits:set,port_detected; sid:2000354; rev:1;)
stats_ports udp any any <> any 3724 (msg:"WoW"; flowbits:set,port_detected; sid:2000355; rev:1;)
stats_ports tcp any any <> any 3784 (msg:"Ventrilo"; flowbits:set,port_detected; sid:2000356; rev:1;)
stats_ports udp any any <> any 3784 (msg:"Ventrilo"; flowbits:set,port_detected; sid:2000357; rev:1;)
stats_ports tcp any any <> any 17500 (msg:"CrazyNet"; flowbits:set,port_detected; sid:2000358; rev:1;)
stats_ports udp any any <> any 17500 (msg:"Ventrilo"; flowbits:set,port_detected; sid:2000359; rev:1;)
stats_ports tcp any any <> any 5353:5354 (msg:"mDNS"; flowbits:set,port_detected; sid:2000360; rev:1;)
stats_ports udp any any <> any 5353:5354 (msg:"mDNS"; flowbits:set,port_detected; sid:2000361; rev:1;)

stats_ports tcp any any <> any 5355 (msg:"LLMNR"; flowbits:set,port_detected; sid:2000362; rev:1;)
stats_ports tcp any any <> any 3544 (msg:"Teredo"; flowbits:set,port_detected; sid:2000364; rev:1;)
stats_ports tcp any any <> any 10201:10204 (msg:"f-protd"; flowbits:set,port_detected; sid:2000370; rev:1;)
stats_ports udp any any <> any 12998:12999 (msg:"SecondLife"; flowbits:set,port_detected; sid:2000371; rev:1;)
stats_ports udp any any <> any 13000:13050 (msg:"SecondLife"; flowbits:set,port_detected; sid:2000372; rev:1;)
stats_ports tcp any any <> any 13195:13196 (msg:"Ontolux"; flowbits:set,port_detected; sid:2000373; rev:1;)
stats_ports udp any any <> any 13195:13196 (msg:"Ontolux"; flowbits:set,port_detected; sid:2000374; rev:1;)
stats_ports tcp any any <> any 1417:1420 (msg:"Timbuktu"; flowbits:set,port_detected; sid:2000375; rev:1;)
stats_ports udp any any <> any 1417:1420 (msg:"Timbuktu"; flowbits:set,port_detected; sid:2000376; rev:1;)
stats_ports tcp any any <> any 1762:1768 (msg:"cft"; flowbits:set,port_detected; sid:2000377; rev:1;)
stats_ports udp any any <> any 1762:1768 (msg:"cft"; flowbits:set,port_detected; sid:2000378; rev:1;)
stats_ports udp any any <> any 1975:1977 (msg:"Cisco TCO"; flowbits:set,port_detected; sid:2000379; rev:1;)
stats_ports tcp any any <> any 2700:2800 (msg:"KnowShowGo"; flowbits:set,port_detected; sid:2000380; rev:1;)
stats_ports udp any any <> any 27000:27006 (msg:"Quake"; flowbits:set,port_detected; sid:2000381; rev:1;)
stats_ports tcp any any <> any 27000:27009 (msg:"FlexNet"; flowbits:set,port_detected; sid:2000382; rev:1;)
stats_ports tcp any any <> any 27500:27900 (msg:"Quake"; flowbits:set,port_detected; sid:2000383; rev:1;)
stats_ports udp any any <> any 27500:27900 (msg:"Quake"; flowbits:set,port_detected; sid:2000384; rev:1;)
stats_ports tcp any any <> any 27900:27901 (msg:"Nintendo"; flowbits:set,port_detected; sid:2000385; rev:1;)
stats_ports udp any any <> any 27900:27901 (msg:"Nintendo"; flowbits:set,port_detected; sid:2000386; rev:1;)
stats_ports tcp any any <> any 27901:27910 (msg:"Quake"; flowbits:set,port_detected; sid:2000387; rev:1;)
stats_ports udp any any <> any 27901:27910 (msg:"Quake"; flowbits:set,port_detected; sid:2000388; rev:1;)
stats_ports tcp any any <> any 27960:27969 (msg:"Quake"; flowbits:set,port_detected; sid:2000389; rev:1;)
stats_ports udp any any <> any 27960:27969 (msg:"Quake"; flowbits:set,port_detected; sid:2000390; rev:1;)
stats_ports tcp any any <> any 29900:29901 (msg:"Nintendo"; flowbits:set,port_detected; sid:2000391; rev:1;)
stats_ports udp any any <> any 29900:29901 (msg:"Nintendo"; flowbits:set,port_detected; sid:2000392; rev:1;)
stats_ports tcp any any <> any 3001:3008 (msg:"Miralix"; flowbits:set,port_detected; sid:2000393; rev:1;)
stats_ports tcp any any <> any 43594:43595 (msg:"RuneScape"; flowbits:set,port_detected; sid:2000394; rev:1;)
stats_ports tcp any any <> any 4433:4436 (msg:"nVision"; flowbits:set,port_detected; sid:2000395; rev:1;)
stats_ports tcp any any <> any 4610:4640 (msg:"QS TestShell"; flowbits:set,port_detected; sid:2000396; rev:1;)
stats_ports tcp any any <> any 5310:5315 (msg:"Ginever.net"; flowbits:set,port_detected; sid:2000397; rev:1;)
stats_ports udp any any <> any 5310:5315 (msg:"Ginever.net"; flowbits:set,port_detected; sid:2000398; rev:1;)
stats_ports tcp any any <> any 6881:7000 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2000401; rev:1;)
stats_ports udp any any <> any 6881:7000 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2000402; rev:1;)
stats_ports tcp any any <> any 7777:7788 (msg:"Unreal"; flowbits:set,port_detected; sid:2000413; rev:1;)
stats_ports udp any any <> any 7777:7788 (msg:"Unreal"; flowbits:set,port_detected; sid:2000414; rev:1;)
stats_ports tcp any any <> any 7787:7788 (msg:"GFI EventsMngr"; flowbits:set,port_detected; sid:2000415; rev:1;)
stats_ports tcp any any <> any 8011:8014 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2000418; rev:1;)

Stats_ports_high_popular.rules

 # Peter Politopoulos, 2011
 # Higher than 1023 but still popular ports in use. From official and un-official sources

stats_ports tcp any any <> any 3868 (msg:"DBP"; flowbits:set,port_detected; sid:2100002; rev:1;)
stats_ports tcp any any <> any 1058 (msg:"nim"; flowbits:set,port_detected; sid:2100004; rev:1;)
stats_ports udp any any <> any 1058 (msg:"nim"; flowbits:set,port_detected; sid:2100005; rev:1;)
stats_ports tcp any any <> any 1059 (msg:"nim"; flowbits:set,port_detected; sid:2100006; rev:1;)
stats_ports udp any any <> any 1059 (msg:"nim"; flowbits:set,port_detected; sid:2100007; rev:1;)
stats_ports tcp any any <> any 1085 (msg:"WebObjects"; flowbits:set,port_detected; sid:2100008; rev:1;)
stats_ports udp any any <> any 1085 (msg:"WebObjects"; flowbits:set,port_detected; sid:2100009; rev:1;)
stats_ports tcp any any <> any 1098 (msg:"rmi"; flowbits:set,port_detected; sid:2100010; rev:1;)
stats_ports tcp any any <> any 1099 (msg:"rmi"; flowbits:set,port_detected; sid:2100012; rev:1;)

stats_ports tcp any any <> any 1140 (msg:"AutoNOC"; flowbits:set,port_detected; sid:2100014; rev:1;)
stats_ports tcp any any <> any 1169 (msg:"Tripwire"; flowbits:set,port_detected; sid:2100016; rev:1;)
stats_ports tcp any any <> any 1182 (msg:"AcceleNet"; flowbits:set,port_detected; sid:2100018; rev:1;)
stats_ports tcp any any <> any 1194 (msg:"OpenVPN"; flowbits:set,port_detected; sid:2100020; rev:1;)
stats_ports udp any any <> any 1194 (msg:"OpenVPN"; flowbits:set,port_detected; sid:2100021; rev:1;)
stats_ports tcp any any <> any 1198 (msg:"cajo"; flowbits:set,port_detected; sid:2100022; rev:1;)
stats_ports tcp any any <> any 1223 (msg:"TGP"; flowbits:set,port_detected; sid:2100024; rev:1;)
stats_ports udp any any <> any 1223 (msg:"TGP"; flowbits:set,port_detected; sid:2100025; rev:1;)
stats_ports tcp any any <> any 1241 (msg:"Nessus"; flowbits:set,port_detected; sid:2100026; rev:1;)
stats_ports tcp any any <> any 1270 (msg:"Ms SCOM"; flowbits:set,port_detected; sid:2100028; rev:1;)
stats_ports udp any any <> any 1270 (msg:"Ms SCOM"; flowbits:set,port_detected; sid:2100029; rev:1;)
stats_ports tcp any any <> any 1293 (msg:"IPSec"; flowbits:set,port_detected; sid:2100030; rev:1;)
stats_ports tcp any any <> any 1387 (msg:"cadsi-lm"; flowbits:set,port_detected; sid:2100032; rev:1;)
stats_ports tcp any any <> any 1434 (msg:"MSSQL"; flowbits:set,port_detected; sid:2100034; rev:1;)
stats_ports tcp any any <> any 1503 (msg:"Win Live Messenger"; flowbits:set,port_detected; sid:2100036; rev:1;)
stats_ports tcp any any <> any 1512 (msg:"WINS"; flowbits:set,port_detected; sid:2100038; rev:1;)
stats_ports tcp any any <> any 1513 (msg:"Garena"; flowbits:set,port_detected; sid:2100040; rev:1;)
stats_ports udp any any <> any 1513 (msg:"Garena"; flowbits:set,port_detected; sid:2100041; rev:1;)
stats_ports tcp any any <> any 1524 (msg:"ingres"; flowbits:set,port_detected; sid:2100042; rev:1;)
stats_ports udp any any <> any 1524 (msg:"ingres"; flowbits:set,port_detected; sid:2100043; rev:1;)
stats_ports tcp any any <> any 1645 (msg:"RADIUS"; flowbits:set,port_detected; sid:2100046; rev:1;)
stats_ports tcp any any <> any 1646 (msg:"RADIUS"; flowbits:set,port_detected; sid:2100048; rev:1;)
stats_ports tcp any any <> any 1677 (msg:"GroupWise"; flowbits:set,port_detected; sid:2100050; rev:1;)
stats_ports udp any any <> any 1677 (msg:"GroupWise"; flowbits:set,port_detected; sid:2100051; rev:1;)
stats_ports tcp any any <> any 1723 (msg:"PPTP"; flowbits:set,port_detected; sid:2100052; rev:1;)
stats_ports tcp any any <> any 1755 (msg:"ms-streaming"; flowbits:set,port_detected; sid:2100054; rev:1;)
stats_ports udp any any <> any 1755 (msg:"ms-streaming"; flowbits:set,port_detected; sid:2100055; rev:1;)
stats_ports tcp any any <> any 1812 (msg:"RADIUS"; flowbits:set,port_detected; sid:2100058; rev:1;)
stats_ports tcp any any <> any 1813 (msg:"RADIUS"; flowbits:set,port_detected; sid:2100060; rev:1;)
stats_ports tcp any any <> any 1883 (msg:"MQTT"; flowbits:set,port_detected; sid:2100062; rev:1;)
stats_ports tcp any any <> any 1947 (msg:"SentinelSRM"; flowbits:set,port_detected; sid:2100064; rev:1;)
stats_ports tcp any any <> any 1970 (msg:"Netop"; flowbits:set,port_detected; sid:2100066; rev:1;)
stats_ports tcp any any <> any 1971 (msg:"Netop"; flowbits:set,port_detected; sid:2100068; rev:1;)
stats_ports tcp any any <> any 1972 (msg:"InterSystems Cache"; flowbits:set,port_detected; sid:2100070; rev:1;)
stats_ports tcp any any <> any 1994 (msg:"Cisco STUN-SDLC"; flowbits:set,port_detected; sid:2100072; rev:1;)
stats_ports tcp any any <> any 1998 (msg:"Cisco XOT"; flowbits:set,port_detected; sid:2100074; rev:1;)
stats_ports tcp any any <> any 2000 (msg:"Cisco SCCP"; flowbits:set,port_detected; sid:2100076; rev:1;)
stats_ports udp any any <> any 2000 (msg:"Cisco SCCP"; flowbits:set,port_detected; sid:2100077; rev:1;)
stats_ports tcp any any <> any 2031 (msg:"mobrien-chat"; flowbits:set,port_detected; sid:2100078; rev:1;)
stats_ports udp any any <> any 2031 (msg:"mobrien-chat"; flowbits:set,port_detected; sid:2100079; rev:1;)
stats_ports tcp any any <> any 2073 (msg:"DataReel db"; flowbits:set,port_detected; sid:2100080; rev:1;)
stats_ports tcp any any <> any 2074 (msg:"Vertel VMF"; flowbits:set,port_detected; sid:2100082; rev:1;)
stats_ports udp any any <> any 2074 (msg:"Vertel VMF"; flowbits:set,port_detected; sid:2100083; rev:1;)
stats_ports tcp any any <> any 2102 (msg:"zephyr"; flowbits:set,port_detected; sid:2100084; rev:1;)
stats_ports tcp any any <> any 2103 (msg:"zephyr"; flowbits:set,port_detected; sid:2100086; rev:1;)
stats_ports tcp any any <> any 2104 (msg:"zephyr"; flowbits:set,port_detected; sid:2100088; rev:1;)
stats_ports tcp any any <> any 2105 (msg:"rlogin"; flowbits:set,port_detected; sid:2100090; rev:1;)
stats_ports tcp any any <> any 2181 (msg:"EForward"; flowbits:set,port_detected; sid:2100092; rev:1;)
stats_ports tcp any any <> any 2210 (msg:"NOAAPORT - MikroTik"; flowbits:set,port_detected; sid:2100094; rev:1;)
stats_ports tcp any any <> any 2211 (msg:"EMWIN - MikroTik"; flowbits:set,port_detected; sid:2100096; rev:1;)
stats_ports tcp any any <> any 2212 (msg:"LeeCO"; flowbits:set,port_detected; sid:2100098; rev:1;)
stats_ports tcp any any <> any 2219 (msg:"NetIQ NCAP"; flowbits:set,port_detected; sid:2100100; rev:1;)

stats_ports tcp any any <> any 2220 (msg:"NetIQ End2End"; flowbits:set,port_detected; sid:2100102; rev:1;)
stats_ports tcp any any <> any 2261 (msg:"CoMotion"; flowbits:set,port_detected; sid:2100104; rev:1;)
stats_ports tcp any any <> any 2262 (msg:"CoMotion"; flowbits:set,port_detected; sid:2100106; rev:1;)
stats_ports tcp any any <> any 2447 (msg:"NNM"; flowbits:set,port_detected; sid:2100108; rev:1;)
stats_ports tcp any any <> any 2483 (msg:"Oracle db"; flowbits:set,port_detected; sid:2100110; rev:1;)
stats_ports tcp any any <> any 2484 (msg:"Oracle db"; flowbits:set,port_detected; sid:2100112; rev:1;)
stats_ports tcp any any <> any 2546 (msg:"EVault"; flowbits:set,port_detected; sid:2100116; rev:1;)
stats_ports tcp any any <> any 2593 (msg:"Ultima Online"; flowbits:set,port_detected; sid:2100118; rev:1;)
stats_ports tcp any any <> any 2612 (msg:"QPasa"; flowbits:set,port_detected; sid:2100120; rev:1;)
stats_ports tcp any any <> any 2713 (msg:"Raven Trinity"; flowbits:set,port_detected; sid:2100122; rev:1;)
stats_ports tcp any any <> any 2714 (msg:"Raven Trinity"; flowbits:set,port_detected; sid:2100124; rev:1;)
stats_ports tcp any any <> any 2735 (msg:"NetIQ"; flowbits:set,port_detected; sid:2100126; rev:1;)
stats_ports tcp any any <> any 2948 (msg:"MMS"; flowbits:set,port_detected; sid:2100132; rev:1;)
stats_ports tcp any any <> any 2949 (msg:"MMS"; flowbits:set,port_detected; sid:2100134; rev:1;)
stats_ports tcp any any <> any 3030 (msg:"NetPanzer"; flowbits:set,port_detected; sid:2100136; rev:1;)
stats_ports tcp any any <> any 3050 (msg:"gds_db"; flowbits:set,port_detected; sid:2100138; rev:1;)
stats_ports tcp any any <> any 3051 (msg:"Galaxy tickets"; flowbits:set,port_detected; sid:2100140; rev:1;)
stats_ports tcp any any <> any 3074 (msg:"Xbox Windows LIVE"; flowbits:set,port_detected; sid:2100142; rev:1;)
stats_ports tcp any any <> any 3225 (msg:"FCIP"; flowbits:set,port_detected; sid:2100144; rev:1;)
stats_ports tcp any any <> any 3233 (msg:"WhiskerControl"; flowbits:set,port_detected; sid:2100146; rev:1;)
stats_ports tcp any any <> any 3235 (msg:"Galaxy tickets"; flowbits:set,port_detected; sid:2100148; rev:1;)
stats_ports tcp any any <> any 3268 (msg:"msft-gc"; flowbits:set,port_detected; sid:2100150; rev:1;)
stats_ports tcp any any <> any 3269 (msg:"msft-gc"; flowbits:set,port_detected; sid:2100152; rev:1;)
stats_ports tcp any any <> any 3300 (msg:"Debate Gopher"; flowbits:set,port_detected; sid:2100154; rev:1;)
stats_ports tcp any any <> any 3305 (msg:"OFTP"; flowbits:set,port_detected; sid:2100156; rev:1;)
stats_ports tcp any any <> any 3306 (msg:"MySQL"; flowbits:set,port_detected; sid:2100158; rev:1;)
stats_ports udp any any <> any 3306 (msg:"MySQL"; flowbits:set,port_detected; sid:2100159; rev:1;)
stats_ports tcp any any <> any 3386 (msg:"GTP CDR"; flowbits:set,port_detected; sid:2100160; rev:1;)
stats_ports tcp any any <> any 3389 (msg:"RDP"; flowbits:set,port_detected; sid:2100162; rev:1;)
stats_ports tcp any any <> any 3396 (msg:"NDPS"; flowbits:set,port_detected; sid:2100164; rev:1;)
stats_ports tcp any any <> any 3412 (msg:"xmlBlaster"; flowbits:set,port_detected; sid:2100166; rev:1;)
stats_ports tcp any any <> any 3455 (msg:"RSVP"; flowbits:set,port_detected; sid:2100168; rev:1;)
stats_ports tcp any any <> any 3478 (msg:"STUN"; flowbits:set,port_detected; sid:2100170; rev:1;)
stats_ports udp any any <> any 3478 (msg:"STUN"; flowbits:set,port_detected; sid:2100171; rev:1;)
stats_ports tcp any any <> any 3516 (msg:"Smartcard"; flowbits:set,port_detected; sid:2100172; rev:1;)
stats_ports tcp any any <> any 3532 (msg:"Raven Remote"; flowbits:set,port_detected; sid:2100174; rev:1;)
stats_ports tcp any any <> any 3606 (msg:"Splitlock"; flowbits:set,port_detected; sid:2100180; rev:1;)
stats_ports tcp any any <> any 3690 (msg:"Subversion"; flowbits:set,port_detected; sid:2100182; rev:1;)
stats_ports tcp any any <> any 3723 (msg:"Battle.net"; flowbits:set,port_detected; sid:2100186; rev:1;)
stats_ports tcp any any <> any 3880 (msg:"IGRS"; flowbits:set,port_detected; sid:2100188; rev:1;)
stats_ports tcp any any <> any 3945 (msg:"EMCADS"; flowbits:set,port_detected; sid:2100190; rev:1;)
stats_ports udp any any <> any 3978 (msg:"OpenTTD"; flowbits:set,port_detected; sid:2100193; rev:1;)
stats_ports udp any any <> any 3979 (msg:"OpenTTD"; flowbits:set,port_detected; sid:2100195; rev:1;)
stats_ports udp any any <> any 4000 (msg:"Diablo"; flowbits:set,port_detected; sid:2100199; rev:1;)
stats_ports tcp any any <> any 4018 (msg:"protocol info"; flowbits:set,port_detected; sid:2100200; rev:1;)
stats_ports tcp any any <> any 4089 (msg:"OpenCORE"; flowbits:set,port_detected; sid:2100202; rev:1;)
stats_ports tcp any any <> any 4093 (msg:"PxPlus"; flowbits:set,port_detected; sid:2100204; rev:1;)
stats_ports tcp any any <> any 4096 (msg:"Timeplex BRE"; flowbits:set,port_detected; sid:2100206; rev:1;)
stats_ports udp any any <> any 4096 (msg:"Timeplex BRE"; flowbits:set,port_detected; sid:2100207; rev:1;)
stats_ports tcp any any <> any 4116 (msg:"Smartcard"; flowbits:set,port_detected; sid:2100208; rev:1;)
stats_ports tcp any any <> any 4172 (msg:"PCoIP"; flowbits:set,port_detected; sid:2100210; rev:1;)
stats_ports udp any any <> any 4172 (msg:"PCoIP"; flowbits:set,port_detected; sid:2100211; rev:1;)

stats_ports tcp any any <> any 4226 (msg:"Aleph One"; flowbits:set,port_detected; sid:2100212; rev:1;)
stats_ports udp any any <> any 4226 (msg:"Aleph One"; flowbits:set,port_detected; sid:2100213; rev:1;)
stats_ports tcp any any <> any 4843 (msg:"OPC UA"; flowbits:set,port_detected; sid:2100216; rev:1;)
stats_ports tcp any any <> any 4894 (msg:"LysKOM"; flowbits:set,port_detected; sid:2100220; rev:1;)
stats_ports tcp any any <> any 4899 (msg:"Radmin"; flowbits:set,port_detected; sid:2100222; rev:1;)
stats_ports udp any any <> any 4899 (msg:"Radmin"; flowbits:set,port_detected; sid:2100223; rev:1;)
stats_ports tcp any any <> any 4950 (msg:"Cylon"; flowbits:set,port_detected; sid:2100224; rev:1;)
stats_ports tcp any any <> any 4982 (msg:"Solar Data"; flowbits:set,port_detected; sid:2100226; rev:1;)
stats_ports udp any any <> any 4982 (msg:"Solar Data"; flowbits:set,port_detected; sid:2100227; rev:1;)
stats_ports tcp any any <> any 4993 (msg:"FTP"; flowbits:set,port_detected; sid:2100228; rev:1;)
stats_ports tcp any any <> any 5001 (msg:"Iperf"; flowbits:set,port_detected; sid:2100230; rev:1;)
stats_ports udp any any <> any 5001 (msg:"Iperf"; flowbits:set,port_detected; sid:2100231; rev:1;)
stats_ports udp any any <> any 5060 (msg:"SIP"; flowbits:set,port_detected; sid:2100237; rev:1;)
stats_ports udp any any <> any 5150 (msg:"ATMP"; flowbits:set,port_detected; sid:2100251; rev:1;)
stats_ports udp any any <> any 5154 (msg:"BZFlag"; flowbits:set,port_detected; sid:2100253; rev:1;)
stats_ports udp any any <> any 5355 (msg:"LLMNR"; flowbits:set,port_detected; sid:2100259; rev:1;)
stats_ports udp any any <> any 5556 (msg:"Freeciv"; flowbits:set,port_detected; sid:2100275; rev:1;)
stats_ports udp any any <> any 5984 (msg:"CouchDB"; flowbits:set,port_detected; sid:2100289; rev:1;)
stats_ports udp any any <> any 6111 (msg:"HP Softbench"; flowbits:set,port_detected; sid:2100293; rev:1;)
stats_ports udp any any <> any 6260 (msg:"eDonkey"; flowbits:set,port_detected; sid:2100295; rev:1;)
stats_ports tcp any any <> any 6346 (msg:"GNUTella"; flowbits:set,port_detected; sid:2100296; rev:1;)
stats_ports udp any any <> any 6346 (msg:"GNUTella"; flowbits:set,port_detected; sid:2100297; rev:1;)
stats_ports udp any any <> any 6347 (msg:"GNUTella"; flowbits:set,port_detected; sid:2100299; rev:1;)
#stats_ports udp any any <> any 6888 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2100311; rev:1;)
stats_ports udp any any <> any 7400 (msg:"RTPS"; flowbits:set,port_detected; sid:2100315; rev:1;)
stats_ports udp any any <> any 7402 (msg:"RTPS"; flowbits:set,port_detected; sid:2100319; rev:1;)
stats_ports tcp any any <> any 8883 (msg:"MQTT"; flowbits:set,port_detected; sid:2100342; rev:1;)
stats_ports tcp any any <> any 9000 (msg:"Various 9000"; flowbits:set,port_detected; sid:2100344; rev:1;)
stats_ports udp any any <> any 9000 (msg:"Various 9000"; flowbits:set,port_detected; sid:2100345; rev:1;)
stats_ports tcp any any <> any 9001 (msg:"Tor"; flowbits:set,port_detected; sid:2100346; rev:1;)
stats_ports udp any any <> any 9001 (msg:"Tor"; flowbits:set,port_detected; sid:2100347; rev:1;)
stats_ports udp any any <> any 9676 (msg:"Spiceworks Desktop"; flowbits:set,port_detected; sid:2100371; rev:1;)
stats_ports tcp any any <> any 9800 (msg:"WebDAV"; flowbits:set,port_detected; sid:2100372; rev:1;)
stats_ports udp any any <> any 9800 (msg:"WebDAV"; flowbits:set,port_detected; sid:2100373; rev:1;)
stats_ports tcp any any <> any 9999 (msg:"eDonkey"; flowbits:set,port_detected; sid:2100376; rev:1;)
stats_ports udp any any <> any 9999 (msg:"eDonkey"; flowbits:set,port_detected; sid:2100377; rev:1;)
stats_ports tcp any any <> any 10000 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100378; rev:1;)
stats_ports udp any any <> any 10000 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100379; rev:1;)
stats_ports tcp any any <> any 10008 (msg:"CROMP"; flowbits:set,port_detected; sid:2100380; rev:1;)
stats_ports udp any any <> any 10008 (msg:"CROMP"; flowbits:set,port_detected; sid:2100381; rev:1;)
stats_ports udp any any <> any 10009 (msg:"Cross Fire"; flowbits:set,port_detected; sid:2100383; rev:1;)
stats_ports udp any any <> any 10050 (msg:"Zabbix"; flowbits:set,port_detected; sid:2100385; rev:1;)
stats_ports tcp any any <> any 10051 (msg:"Zabbix"; flowbits:set,port_detected; sid:2100386; rev:1;)
stats_ports udp any any <> any 10115 (msg:"NetIQ"; flowbits:set,port_detected; sid:2100393; rev:1;)
stats_ports udp any any <> any 12345 (msg:"NetBus(ter)"; flowbits:set,port_detected; sid:2100399; rev:1;)
stats_ports udp any any <> any 15000 (msg:"Various 15000"; flowbits:set,port_detected; sid:2100417; rev:1;)
stats_ports udp any any <> any 15345 (msg:"XPilot"; flowbits:set,port_detected; sid:2100419; rev:1;)
stats_ports udp any any <> any 18301 (msg:"Audition Online"; flowbits:set,port_detected; sid:2100429; rev:1;)
stats_ports udp any any <> any 18401 (msg:"Audition Online"; flowbits:set,port_detected; sid:2100435; rev:1;)
stats_ports tcp any any <> any 20000 (msg:"DNP - Usermin"; flowbits:set,port_detected; sid:2100454; rev:1;)
stats_ports udp any any <> any 20000 (msg:"DNP - Usermin"; flowbits:set,port_detected; sid:2100455; rev:1;)
stats_ports udp any any <> any 25565 (msg:"Minecraft - MySQL"; flowbits:set,port_detected; sid:2100465; rev:1;)

stats_ports tcp any any <> any 40000 (msg:"Rt Industr Ethernet"; flowbits:set,port_detected; sid:2100470; rev:1;)

stats_ports udp any any <> any 40000 (msg:"Rt Industr Ethernet"; flowbits:set,port_detected; sid:2100471; rev:1;)

stats_ports udp any any <> any 40001 (msg:"Rt Industr Ethernet"; flowbits:set,port_detected; sid:2100473; rev:1;)

stats_ports udp any any <> any 5004 (msg:"RTP"; flowbits:set,port_detected; sid:2100479; rev:1;)

stats_ports tcp any any <> any 1337 (msg:"Various P2P Encrypted FileSystems"; flowbits:set,port_detected; sid:2100482; rev:1;)

stats_ports udp any any <> any 1337 (msg:"Various P2P Encrypted FileSystems"; flowbits:set,port_detected; sid:2100483; rev:1;)

stats_ports tcp any any <> any 1761 (msg:"cft-0"; flowbits:set,port_detected; sid:2100484; rev:1;)

stats_ports udp any any <> any 1761 (msg:"cft-0"; flowbits:set,port_detected; sid:2100485; rev:1;)

stats_ports tcp any any <> any 2710 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2100486; rev:1;)

stats_ports udp any any <> any 2710 (msg:"BitTorrent"; flowbits:set,port_detected; sid:2100487; rev:1;)

stats_ports tcp any any <> any 1707 (msg:"L2TP"; flowbits:set,port_detected; sid:2100488; rev:1;)

stats_ports tcp any any <> any 1025 (msg:"NFS - IIS"; flowbits:set,port_detected; sid:2100489; rev:1;)

stats_ports tcp any any <> any 1029 (msg:"DCOM"; flowbits:set,port_detected; sid:2100491; rev:1;)

stats_ports tcp any any <> any 1080 (msg:"SOCKS proxy"; flowbits:set,port_detected; sid:2100492; rev:1;)

stats_ports tcp any any <> any 1109 (msg:"KPOP"; flowbits:set,port_detected; sid:2100493; rev:1;)

stats_ports tcp any any <> any 1176 (msg:"Indigo Home auto"; flowbits:set,port_detected; sid:2100494; rev:1;)

stats_ports tcp any any <> any 1200 (msg:"scol"; flowbits:set,port_detected; sid:2100495; rev:1;)

stats_ports tcp any any <> any 1214 (msg:"Kazaa"; flowbits:set,port_detected; sid:2100496; rev:1;)

stats_ports tcp any any <> any 1217 (msg:"Uvora Online"; flowbits:set,port_detected; sid:2100497; rev:1;)

stats_ports tcp any any <> any 1220 (msg:"QuickTime"; flowbits:set,port_detected; sid:2100498; rev:1;)

stats_ports tcp any any <> any 1236 (msg:"BindView"; flowbits:set,port_detected; sid:2100499; rev:1;)

stats_ports tcp any any <> any 1301 (msg:"OBDNet"; flowbits:set,port_detected; sid:2100500; rev:1;)

stats_ports tcp any any <> any 1309 (msg:"jtagd"; flowbits:set,port_detected; sid:2100501; rev:1;)

stats_ports tcp any any <> any 1311 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100502; rev:1;)

stats_ports tcp any any <> any 1313 (msg:"Xbiim"; flowbits:set,port_detected; sid:2100503; rev:1;)

stats_ports tcp any any <> any 1352 (msg:"IBM Lotus Notes"; flowbits:set,port_detected; sid:2100504; rev:1;)

stats_ports tcp any any <> any 1414 (msg:"IBM WebSphere"; flowbits:set,port_detected; sid:2100505; rev:1;)

stats_ports tcp any any <> any 1431 (msg:"RGTP"; flowbits:set,port_detected; sid:2100506; rev:1;)

stats_ports tcp any any <> any 1433 (msg:"MSSQL"; flowbits:set,port_detected; sid:2100507; rev:1;)

stats_ports tcp any any <> any 1470 (msg:"Kiwi"; flowbits:set,port_detected; sid:2100508; rev:1;)

stats_ports tcp any any <> any 1494 (msg:"XenApp"; flowbits:set,port_detected; sid:2100509; rev:1;)

stats_ports tcp any any <> any 1500 (msg:"GuardianPro"; flowbits:set,port_detected; sid:2100510; rev:1;)

stats_ports tcp any any <> any 1521 (msg:"nCube LM - Old Oracle"; flowbits:set,port_detected; sid:2100511; rev:1;)

stats_ports tcp any any <> any 1526 (msg:"Oracle db"; flowbits:set,port_detected; sid:2100512; rev:1;)

stats_ports tcp any any <> any 1533 (msg:"MSSQL"; flowbits:set,port_detected; sid:2100513; rev:1;)

stats_ports tcp any any <> any 1666 (msg:"Perforce"; flowbits:set,port_detected; sid:2100514; rev:1;)

stats_ports tcp any any <> any 1688 (msg:"MS Activation"; flowbits:set,port_detected; sid:2100515; rev:1;)

stats_ports tcp any any <> any 1716 (msg:"Americas Army"; flowbits:set,port_detected; sid:2100516; rev:1;)

stats_ports tcp any any <> any 1720 (msg:"H.323"; flowbits:set,port_detected; sid:2100517; rev:1;)

stats_ports tcp any any <> any 1863 (msg:"Win Live Messenger"; flowbits:set,port_detected; sid:2100518; rev:1;)

stats_ports tcp any any <> any 1886 (msg:"Leonardo"; flowbits:set,port_detected; sid:2100519; rev:1;)

stats_ports tcp any any <> any 1920 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100520; rev:1;)

stats_ports tcp any any <> any 1935 (msg:"RTMP"; flowbits:set,port_detected; sid:2100521; rev:1;)

stats_ports tcp any any <> any 1984 (msg:"Big Brother"; flowbits:set,port_detected; sid:2100522; rev:1;)

stats_ports tcp any any <> any 1997 (msg:"Chizmo Net"; flowbits:set,port_detected; sid:2100523; rev:1;)

stats_ports tcp any any <> any 2002 (msg:"ACS"; flowbits:set,port_detected; sid:2100524; rev:1;)

stats_ports tcp any any <> any 2041 (msg:"Mail.Ru"; flowbits:set,port_detected; sid:2100525; rev:1;)

stats_ports tcp any any <> any 2053 (msg:"Kerberos"; flowbits:set,port_detected; sid:2100526; rev:1;)

stats_ports tcp any any <> any 2082 (msg:"CPanel"; flowbits:set,port_detected; sid:2100527; rev:1;)

stats_ports tcp any any <> any 2083 (msg:"CPanel - Radsec"; flowbits:set,port_detected; sid:2100528; rev:1;)

stats_ports tcp any any <> any 2086 (msg:"GNUnet - WebHost Manager"; flowbits:set,port_detected; sid:2100529; rev:1;)

stats_ports tcp any any <> any 2087 (msg:"WebHost Manager"; flowbits:set,port_detected; sid:2100530; rev:1;)
stats_ports tcp any any <> any 2095 (msg:"CPanel"; flowbits:set,port_detected; sid:2100531; rev:1;)
stats_ports tcp any any <> any 2096 (msg:"CPanel"; flowbits:set,port_detected; sid:2100532; rev:1;)
stats_ports tcp any any <> any 2144 (msg:"LiveVault"; flowbits:set,port_detected; sid:2100533; rev:1;)
stats_ports tcp any any <> any 2145 (msg:"LiveVault"; flowbits:set,port_detected; sid:2100534; rev:1;)
stats_ports tcp any any <> any 2161 (msg:"APC Agent"; flowbits:set,port_detected; sid:2100535; rev:1;)
stats_ports tcp any any <> any 2221 (msg:"ESET AV"; flowbits:set,port_detected; sid:2100536; rev:1;)
stats_ports tcp any any <> any 2222 (msg:"ESET AV"; flowbits:set,port_detected; sid:2100537; rev:1;)
stats_ports tcp any any <> any 2301 (msg:"HP Sys Mng"; flowbits:set,port_detected; sid:2100538; rev:1;)
stats_ports tcp any any <> any 2369 (msg:"BMC Sw Ctrl"; flowbits:set,port_detected; sid:2100539; rev:1;)
stats_ports tcp any any <> any 2370 (msg:"BMC Sw Ctrl"; flowbits:set,port_detected; sid:2100540; rev:1;)
stats_ports tcp any any <> any 2381 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100541; rev:1;)
stats_ports tcp any any <> any 2401 (msg:"CVS"; flowbits:set,port_detected; sid:2100542; rev:1;)
stats_ports tcp any any <> any 2404 (msg:"IEC 60870-5-104"; flowbits:set,port_detected; sid:2100543; rev:1;)
stats_ports tcp any any <> any 2500 (msg:"TheosNet"; flowbits:set,port_detected; sid:2100544; rev:1;)
stats_ports tcp any any <> any 2501 (msg:"TheosNet"; flowbits:set,port_detected; sid:2100545; rev:1;)
stats_ports tcp any any <> any 2525 (msg:"SMTP"; flowbits:set,port_detected; sid:2100546; rev:1;)
stats_ports tcp any any <> any 2535 (msg:"MADCAP"; flowbits:set,port_detected; sid:2100547; rev:1;)
stats_ports tcp any any <> any 2598 (msg:"ICA"; flowbits:set,port_detected; sid:2100548; rev:1;)
stats_ports tcp any any <> any 2599 (msg:"SonicWALL"; flowbits:set,port_detected; sid:2100549; rev:1;)
stats_ports tcp any any <> any 2610 (msg:"Dark Ages"; flowbits:set,port_detected; sid:2100550; rev:1;)
stats_ports tcp any any <> any 2638 (msg:"Sybase db"; flowbits:set,port_detected; sid:2100551; rev:1;)
stats_ports tcp any any <> any 2947 (msg:"gpsd GPS"; flowbits:set,port_detected; sid:2100552; rev:1;)
stats_ports tcp any any <> any 2967 (msg:"Symantec AV"; flowbits:set,port_detected; sid:2100553; rev:1;)
stats_ports tcp any any <> any 3000 (msg:"Miralix - Cloud9"; flowbits:set,port_detected; sid:2100554; rev:1;)
stats_ports tcp any any <> any 3017 (msg:"Miralix"; flowbits:set,port_detected; sid:2100555; rev:1;)
stats_ports tcp any any <> any 3025 (msg:"netpd.org"; flowbits:set,port_detected; sid:2100556; rev:1;)
stats_ports tcp any any <> any 3100 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100557; rev:1;)
stats_ports tcp any any <> any 3101 (msg:"BlackBerry"; flowbits:set,port_detected; sid:2100558; rev:1;)
stats_ports tcp any any <> any 3128 (msg:"HTTP cache - Squid"; flowbits:set,port_detected; sid:2100559; rev:1;)
stats_ports tcp any any <> any 3260 (msg:"iSCSI"; flowbits:set,port_detected; sid:2100560; rev:1;)
stats_ports tcp any any <> any 3299 (msg:"SAP"; flowbits:set,port_detected; sid:2100562; rev:1;)
stats_ports tcp any any <> any 3313 (msg:"Verisys"; flowbits:set,port_detected; sid:2100563; rev:1;)
stats_ports tcp any any <> any 3333 (msg:"Caller ID"; flowbits:set,port_detected; sid:2100564; rev:1;)
stats_ports tcp any any <> any 3423 (msg:"xTrm"; flowbits:set,port_detected; sid:2100565; rev:1;)
stats_ports tcp any any <> any 3424 (msg:"xTrm"; flowbits:set,port_detected; sid:2100566; rev:1;)
stats_ports tcp any any <> any 3483 (msg:"SlimProto"; flowbits:set,port_detected; sid:2100567; rev:1;)
stats_ports tcp any any <> any 3535 (msg:"SMTP"; flowbits:set,port_detected; sid:2100568; rev:1;)
stats_ports tcp any any <> any 3632 (msg:"distributed compiler"; flowbits:set,port_detected; sid:2100569; rev:1;)
stats_ports tcp any any <> any 3689 (msg:"DAAP"; flowbits:set,port_detected; sid:2100570; rev:1;)
stats_ports tcp any any <> any 3800 (msg:"HGG"; flowbits:set,port_detected; sid:2100571; rev:1;)
stats_ports tcp any any <> any 3872 (msg:"Oracle db"; flowbits:set,port_detected; sid:2100572; rev:1;)
stats_ports tcp any any <> any 3899 (msg:"Remote Admin"; flowbits:set,port_detected; sid:2100573; rev:1;)
stats_ports tcp any any <> any 3900 (msg:"udt_os"; flowbits:set,port_detected; sid:2100574; rev:1;)
stats_ports tcp any any <> any 4007 (msg:"PrintBuzzer"; flowbits:set,port_detected; sid:2100576; rev:1;)
stats_ports tcp any any <> any 4111 (msg:"Xgrid"; flowbits:set,port_detected; sid:2100577; rev:1;)
stats_ports tcp any any <> any 4201 (msg:"MUD"; flowbits:set,port_detected; sid:2100579; rev:1;)
stats_ports tcp any any <> any 4321 (msg:"RWhois"; flowbits:set,port_detected; sid:2100581; rev:1;)
stats_ports tcp any any <> any 4567 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100582; rev:1;)
stats_ports tcp any any <> any 4664 (msg:"Google Desktop"; flowbits:set,port_detected; sid:2100584; rev:1;)
stats_ports tcp any any <> any 4711 (msg:"eMule"; flowbits:set,port_detected; sid:2100585; rev:1;)
stats_ports tcp any any <> any 4728 (msg:"DMP"; flowbits:set,port_detected; sid:2100587; rev:1;)

stats_ports tcp any any <> any 4747 (msg:"Apprentice"; flowbits:set,port_detected; sid:2100588; rev:1;)
stats_ports tcp any any <> any 4750 (msg:"BladeLogic"; flowbits:set,port_detected; sid:2100589; rev:1;)
stats_ports tcp any any <> any 5000 (msg:"UPnP - VTun VPN"; flowbits:set,port_detected; sid:2100591; rev:1;)
stats_ports tcp any any <> any 5050 (msg:"Yahoo Msg"; flowbits:set,port_detected; sid:2100594; rev:1;)
stats_ports tcp any any <> any 5190 (msg:"ICQ - AOL"; flowbits:set,port_detected; sid:2100606; rev:1;)
stats_ports tcp any any <> any 5222 (msg:"XMPP"; flowbits:set,port_detected; sid:2100607; rev:1;)
stats_ports tcp any any <> any 5555 (msg:"SAP"; flowbits:set,port_detected; sid:2100620; rev:1;)
stats_ports tcp any any <> any 7001 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100657; rev:1;)
stats_ports tcp any any <> any 7777 (msg:"iChat - OCFS2"; flowbits:set,port_detected; sid:2100673; rev:1;)
stats_ports tcp any any <> any 7778 (msg:"MUD"; flowbits:set,port_detected; sid:2100674; rev:1;)
stats_ports tcp any any <> any 8000 (msg:"SHOUTcast"; flowbits:set,port_detected; sid:2100678; rev:1;)
stats_ports tcp any any <> any 8002 (msg:"Cisco CM"; flowbits:set,port_detected; sid:2100680; rev:1;)
stats_ports tcp any any <> any 8008 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100681; rev:1;)
stats_ports tcp any any <> any 8080 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100685; rev:1;)
stats_ports tcp any any <> any 8200 (msg:"GoToMyPC"; flowbits:set,port_detected; sid:2100697; rev:1;)
stats_ports tcp any any <> any 8887 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100708; rev:1;)
stats_ports tcp any any <> any 9030 (msg:"Tor"; flowbits:set,port_detected; sid:2100712; rev:1;)
stats_ports tcp any any <> any 9090 (msg:"HTTP - Web"; flowbits:set,port_detected; sid:2100718; rev:1;)
stats_ports tcp any any <> any 9100 (msg:"PDL"; flowbits:set,port_detected; sid:2100720; rev:1;)
stats_ports tcp any any <> any 9191 (msg:"PocketMoney"; flowbits:set,port_detected; sid:2100721; rev:1;)
stats_ports tcp any any <> any 10001 (msg:"Lantronix UDS-10"; flowbits:set,port_detected; sid:2100731; rev:1;)
stats_ports tcp any any <> any 10025 (msg:"smtp"; flowbits:set,port_detected; sid:2100734; rev:1;)
stats_ports tcp any any <> any 10200 (msg:"fpscand"; flowbits:set,port_detected; sid:2100735; rev:1;)
stats_ports tcp any any <> any 19880 (msg:"Softros LAN Msg"; flowbits:set,port_detected; sid:2100755; rev:1;)
stats_ports tcp any any <> any 21001 (msg:"AMLFilter"; flowbits:set,port_detected; sid:2100758; rev:1;)
stats_ports tcp any any <> any 30564 (msg:"Multiplicity"; flowbits:set,port_detected; sid:2100774; rev:1;)
stats_ports tcp any any <> any 1319 (msg:"AMX ICSP"; flowbits:set,port_detected; sid:2100789; rev:1;)
stats_ports udp any any <> any 1200 (msg:"Steam"; flowbits:set,port_detected; sid:2100795; rev:1;)
stats_ports udp any any <> any 1234 (msg:"VLC"; flowbits:set,port_detected; sid:2100796; rev:1;)
stats_ports udp any any <> any 1501 (msg:"NG GuardianPro"; flowbits:set,port_detected; sid:2100797; rev:1;)
stats_ports udp any any <> any 1719 (msg:"H.323"; flowbits:set,port_detected; sid:2100801; rev:1;)
stats_ports udp any any <> any 1900 (msg:"UPnP"; flowbits:set,port_detected; sid:2100803; rev:1;)
stats_ports udp any any <> any 1967 (msg:"Cisco IP SLAs"; flowbits:set,port_detected; sid:2100804; rev:1;)
stats_ports udp any any <> any 1985 (msg:"Cisco HSRP"; flowbits:set,port_detected; sid:2100805; rev:1;)
stats_ports udp any any <> any 2302 (msg:"Halo"; flowbits:set,port_detected; sid:2100814; rev:1;)
stats_ports udp any any <> any 2945 (msg:"H.248"; flowbits:set,port_detected; sid:2100820; rev:1;)
stats_ports udp any any <> any 3000 (msg:"DIS"; flowbits:set,port_detected; sid:2100821; rev:1;)
stats_ports udp any any <> any 3544 (msg:"Teredo"; flowbits:set,port_detected; sid:2100824; rev:1;)
stats_ports udp any any <> any 4672 (msg:"eMule"; flowbits:set,port_detected; sid:2100831; rev:1;)
stats_ports udp any any <> any 5000 (msg:"FlightGear - VTun VPN"; flowbits:set,port_detected; sid:2100832; rev:1;)
stats_ports udp any any <> any 5678 (msg:"MNDP"; flowbits:set,port_detected; sid:2100841; rev:1;)
stats_ports udp any any <> any 6001 (msg:"X11"; flowbits:set,port_detected; sid:2100842; rev:1;)
stats_ports udp any any <> any 6112 (msg:"dtspcd"; flowbits:set,port_detected; sid:2100843; rev:1;)
stats_ports udp any any <> any 6257 (msg:"WinMX"; flowbits:set,port_detected; sid:2100844; rev:1;)
stats_ports udp any any <> any 6343 (msg:"SFlow"; flowbits:set,port_detected; sid:2100845; rev:1;)
stats_ports udp any any <> any 6679 (msg:"OSAUT"; flowbits:set,port_detected; sid:2100849; rev:1;)
stats_ports udp any any <> any 8701 (msg:"SoftPerfect Bw Mngr"; flowbits:set,port_detected; sid:2100857; rev:1;)
stats_ports udp any any <> any 8767 (msg:"TeamSpeak"; flowbits:set,port_detected; sid:2100859; rev:1;)
stats_ports udp any any <> any 8768 (msg:"TeamSpeak"; flowbits:set,port_detected; sid:2100860; rev:1;)
stats_ports udp any any <> any 8888 (msg:"NewsEDGE"; flowbits:set,port_detected; sid:2100861; rev:1;)
stats_ports udp any any <> any 9600 (msg:"Omron FINS"; flowbits:set,port_detected; sid:2100865; rev:1;)
stats_ports udp any any <> any 16384 (msg:"Iron Mountain"; flowbits:set,port_detected; sid:2100876; rev:1;)

stats_ports tcp any any <> any 1550 (msg:"Gadu-Gadu"; flowbits:set,port_detected; sid:2100886; rev:1;)

stats_ports tcp any any <> any 1627 (msg:"iSketch"; flowbits:set,port_detected; sid:2100888; rev:1;)

stats_ports tcp any any <> any 2030 (msg:"Oracle"; flowbits:set,port_detected; sid:2100890; rev:1;)

stats_ports udp any any <> any 2030 (msg:"Oracle"; flowbits:set,port_detected; sid:2100891; rev:1;)

stats_ports tcp any any <> any 4100 (msg:"WatchGuard"; flowbits:set,port_detected; sid:2100892; rev:1;)

stats_ports udp any any <> any 4100 (msg:"WatchGuard"; flowbits:set,port_detected; sid:2100893; rev:1;)

stats_ports tcp any any <> any 6571 (msg:"Win Live Messenger"; flowbits:set,port_detected; sid:2100894; rev:1;)

stats_ports udp any any <> any 6571 (msg:"Win Live Messenger"; flowbits:set,port_detected; sid:2100895; rev:1;)

stats_ports udp any any <> any 10017 (msg:"HPUXrexd"; flowbits:set,port_detected; sid:2100897; rev:1;)

stats_ports udp any any <> any 10480 (msg:"SWAT4"; flowbits:set,port_detected; sid:2100901; rev:1;)

stats_ports udp any any <> any 11211 (msg:"memcached"; flowbits:set,port_detected; sid:2100903; rev:1;)

stats_ports tcp any any <> any 11294 (msg:"Blood Quest"; flowbits:set,port_detected; sid:2100906; rev:1;)

stats_ports udp any any <> any 11294 (msg:"Blood Quest"; flowbits:set,port_detected; sid:2100907; rev:1;)

stats_ports udp any any <> any 11371 (msg:"OpenPGP"; flowbits:set,port_detected; sid:2100909; rev:1;)

stats_ports udp any any <> any 11576 (msg:"IPStor"; flowbits:set,port_detected; sid:2100911; rev:1;)

stats_ports udp any any <> any 19999 (msg:"DNP"; flowbits:set,port_detected; sid:2100913; rev:1;)

stats_ports udp any any <> any 24800 (msg:"Synergy"; flowbits:set,port_detected; sid:2100923; rev:1;)

stats_ports tcp any any <> any 27015 (msg:"Source engine"; flowbits:set,port_detected; sid:2100930; rev:1;)

stats_ports udp any any <> any 27015 (msg:"Source engine"; flowbits:set,port_detected; sid:2100931; rev:1;)

stats_ports udp any any <> any 27017 (msg:"mongoDB"; flowbits:set,port_detected; sid:2100935; rev:1;)

stats_ports udp any any <> any 28000 (msg:"Bitfighter"; flowbits:set,port_detected; sid:2100939; rev:1;)

stats_ports udp any any <> any 29920 (msg:"Nintendo"; flowbits:set,port_detected; sid:2100947; rev:1;)

stats_ports udp any any <> any 30000 (msg:"Pokemon"; flowbits:set,port_detected; sid:2100949; rev:1;)

Stats_undetected.rules

 # 2011 Peter Politopoulos

 # This file matches the packets that are not part of a known flow-type (yet)

stats_undetected tcp any any <> any any (msg:"UNKNOWN TCP"; sid:1300000;flowbits:isnotset,port_detected; flowbits:isnotset,tracker_detected; rev:1;)

stats_undetected udp any any <> any any (msg:"UNKNOWN UDP"; sid:1300001;flowbits:isnotset,port_detected; flowbits:isnotset,tracker_detected; rev:1;)

Stats.rules

 # Peter Politopoulos, 2011

 # This file contains the rules matching non-tcp/udp packets

stats icmp any any <> any any (msg:"ICMP"; sid:1000000; rev:1;)

Bibliography
and links

[1] ITU Statistics. Available from:

http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf

[2] M. Roesch. Snort: Lightweight intrusion detection for networks. In Proceedings of the 1999 USENIX LISA Systems Administration Conference, November 1999.

[3] V. Paxson. Bro: A system for detecting network intruders in real-time. In Proceedings of the 7th USENIX Security Symposium, January 1998.

[4] openinfosecfoundation. What is Suricata. Available from:

https://redmine.openinfosecfoundation.org/projects/suricata/wiki/What_is_Suricata

[5] A. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. Passive And Active Network Measurement: 6th International Workshop, PAM 2005, Boston, MA, USA, March 31-April 1, 2005: Proceedings, 2005.

[6] B. Cohen. BitTorrent protocol specification. First Workshop on Economics of Peer-to-Peer Systems (P2P'03).

[7] Salman Baset and Henning Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. Computing Research Repository, 0412017, 2004.

[8] rrdtool. http://oss.oetiker.ch/rrdtool/.

[9] S. McCanne, C. Leres, and V. Jacobson. libpcap. Lawrence Berkeley Laboratory, Berkeley, CA. (software available from http://www.tcpdump.org/).

[10] Judy Novak, Steve Sturges. Target-Based TCP Stream Reassembly.

http://www.Snort.org/docs/stream5-modelAug032007.pdf

[11] Thomas Ptacek and Timothy Newsham. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Secure Networks Whitepaper, August 1998.

[12] Alfred V. Aho and Margaret J. Corasick. 1975. Efficient string matching: an aid to bibliographic search. Commun. ACM 18, 6 (June 1975), 333-340.

[13] S. Antonatos, K.G. Anagnostakis, E. Markatos. and M. Polychronakis. Performance analysis of content matching intrusion detection systems: International Symposium on Applications and the Internet, 2004. Proceedings, 2004.

[14] Snort User's Manual. Available Online:

http://www.Snort.org/docs

[15] IANA official port-number assignments. Available Online:

http://www.iana.org/assignments/port-numbers

[16] SANS Institute, Internet Storm Center. Available Online:

http://isc.sans.edu/port.html

[17] SpeedGuide Port Database. Available Online:

http://www.speedguide.net/ports.php

[18] OpenWRT GNU/Linux for Home Routers. Available Online:

https://www.openwrt.org/

[19] University of Crete, Greece

https://www.uoc.gr/

[20] GrNet, Network for Research and Education, Greece

https://www.grnet.gr/

[21] Distributed Computing Systems Laboratory of the Foundation for Research and Technology, Crete, Greece.

https://www.dcs.ics.forth.gr/

[22] PCRE - Perl Compatible Regular Expressions. Available Online:

http://www.pcre.org/

[23] OpenDPI - Open Source version of ipoque's DPI engine. Available Online:

http://www.opendpi.org/

[24] A. Dainotti, W. de Donato, A. Pescapè, "TIE: a Community-Oriented Traffic Classification Platform", International Workshop on Traffic Monitoring and Analysis (TMA'09) @ IFIP Networking 2009 - May 2009, Aachen (Germany)

[25] D. Antoniades, M. Polychronakis, S. Antonatos, E. P. Markatos, S. Ubik, and A. Oslebo. Appmon: An application for accurate per application traffic characterization. In Proceedings of IST Broadband Europe 2006 Conference, December 2006.

[26] LOBSTER - pilot European infrastructure for accurate Internet traffic monitoring. Available Online:

http://www.ist-lobster.org/

[27] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "Kiss: Stochastic packet inspection classifier for udp traffic", IEEE Transactions on Networking, 12(4), 2010.

[28] Hjelmvik, E., John, W.: Breaking and improving protocol obfuscation. Tech. Rep. 2010-05, Computer Science and Engineering, Chalmers University of Technology, 2010.

Available Online:

http://www.iis.se/docs/hjelmvik_breaking.pdf

[29] S. Ortiz Jr. Is Peer to Peer on the Decline? IEEE Computer Society, 2011, Technology News, p. 11.

[30] I. U. Haq, S. Ali, H. Khan, and S. A. Khayam. What is the impact of p2p traffic on anomaly detection? In Proceedings of the 13th international conference on Recent advances in intrusion detection, RAID'10, pages 1–17, 2010.

[31] Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, J. Ucles. HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In Proceedings of IEEE Workshop on Information Assurance and Security, pages 85-90, 2001.

[32] Facebook statistics. Available online:

http://www.facebook.com/press/info.php?statistics

[33] Reuters report on Anonymous hacker group. Available online:

http://www.reuters.com/article/2011/07/29/us-cyber-mantech-idUSTRE76S6IB20110729

[34] BBC report on internet attack rise. Available online:

http://www.bbc.co.uk/news/technology-13122339