Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 6, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

The purpose of this document is to define a series of test scenarios which may be used to generate statistics that should help to better understand the performance of network devices under realistic loading.
conditions. Additionally, this document provides suggestions on which statistics may be the most useful for determining network device performance under realistic deployment scenarios.

Table of Contents

1. Introduction ........................................ 4
   1.1. Requirements Language .......................... 4
2. Scope ............................................. 4
3. Test Setup .......................................... 5
   3.1. Test Considerations ............................ 5
   3.2. Clients and Servers ............................ 5
   3.3. Traffic Generation Requirements ................. 6
   3.4. Multiple Client/Server Testing .................. 6
   3.5. Network Address Translation ..................... 6
   3.6. TCP Stack Considerations ....................... 7
   3.7. Other Considerations ........................... 7
4. Benchmarking Tests ................................ 7
   4.1. Maximum Application Connection Establishment Rate . 7
       4.1.1. Objective ................................... 7
       4.1.2. Setup Parameters ........................... 7
           4.1.2.1. Transport-Layer Parameters ............ 7
           4.1.2.2. Application-Layer Parameters .......... 7
       4.1.3. Procedure ................................... 8
       4.1.4. Measurement ................................ 8
           4.1.4.1. Maximum Application Session Establishment Rate . 8
           4.1.4.2. Application Session Setup Time .......... 8
           4.1.4.3. Application Session Response Time ...... 8
           4.1.4.4. Application Session Time To Close ...... 8
           4.1.4.5. Application Latency ...................... 8
       4.2. Application Throughput ......................... 9
           4.2.1. Objective .................................. 9
           4.2.2. Setup Parameters .......................... 9
               4.2.2.1. Parameters .......................... 9
           4.2.3. Procedure ................................ 9
           4.2.4. Measurement ................................ 9
               4.2.4.1. Maximum Throughput .................... 9
               4.2.4.2. Packet Loss ........................... 9
               4.2.4.3. Application Setup Time ................. 9
               4.2.4.4. Application Response Time .............. 9
               4.2.4.5. Application Session Time To Close .... 10
               4.2.4.6. Application Latency .................... 10
       4.3. Denial of Service Attack Mitigation ............. 10
           4.3.1. Objective ................................ 10
           4.3.2. Setup Parameters .......................... 10
           4.3.3. Procedure ................................ 10
           4.3.4. Measurement ................................ 11
1. Introduction

The purpose of this Internet Draft is to define and provide a set of benchmarks useful for evaluating content-aware network devices. As processing resources have become faster and cheaper, network devices now utilize information far deeper inside the network packet than ever before. No longer are devices looking simply at TCP/IP header information and bits of application headers; devices are now decoding application layer protocols and inspecting them for conformance to a given rule set, anomalies and even security signatures. These devices have commonly become known as content-aware.

Many of the terms used throughout this draft have previously been defined in "Benchmarking Terminology for Firewall Performance" RFC 2647 [1]. This document SHOULD be consulted prior to using this document. The Benchmarking Methodology Working Group (BMWG) has previously defined methodologies for network interconnect devices with RFC 2544 [2] and firewall performance with RFC 3511 [3]. This draft seeks to enhance these methodologies to provide even more realistic results.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

2. Scope

Content-aware devices take many forms, shapes and architectures. These devices are advanced network interconnect devices that inspect deep into the application payload of network data packets to do classification. They may be as simple as a firewall that uses application data inspection for rule set enforcement, or they may have advanced functionality such as doing protocol decoding and validation, anti-virus, anti-spam and even application exploit filtering.

This document is strictly focused on examining performance and robustness across a wide range of metrics that will help to predict device performance when deployed in live networks. These metrics will be, wherever possible, implementation independent.

It should also be noted that the purpose of this document is not to define functional testing of the potential features in the Device/System Under Test (DUT/SUT)[1] nor specify the configurations that should be tested. Various definitions of proper operation and
configuration may be appropriate for different deployments, thus those parameters are outside the scope of this document.

3. Test Setup

This document will be applicable to most test configurations and will not be confined to a discussion on specific test configurations. Since each DUT/SUT will have their own unique configuration, users MUST configure their device with the same parameters that would be used in a live deployment of the device.

The lines between network boundaries are rapidly blurring. Every port on a device could be content-aware when using a fully meshed network topology. Organizations deploying content-aware devices are doing so throughout their network infrastructure. These devices inspect deep into the application flow to perform quality of service monitoring, filtering, metering, threat mitigation and more.

Figure 1 illustrates a network topology that is fully meshed.

```
+----------+    +----------+    +----------+
|          |    |          |    |          |
| Servers/ |____|  DUT    |____| Servers/ |
| Clients  |    |          |    | Clients  |
+----------+    +----------+    +----------+

Fully Meshed Device
```

Figure 1: Fully Meshed Device

This document will also apply to the network configurations specified by Figures 1 and 2 in RFC 3511.

3.1. Test Considerations

3.2. Clients and Servers

Content-aware device testing SHOULD involve multiple clients and multiple servers. As with RFC 3511 [3], this methodology will use the terms virtual clients/servers throughout. Similarly defined in RFC 3511 [3], a data source may emulate multiple clients and/or servers within the context of the same test scenario. The test report MUST indicate the number of virtual clients/servers used during the test. In Appendix C of RFC 2544 [2], the range of IP addresses assigned to the BMWG by the IANA are listed. This address range SHOULD be adhered to in accordance with RFC 2544 [2].
3.3. Traffic Generation Requirements

The explicit purposes of content-aware devices vary widely, but most of these devices use information deeper inside the application flow to make decisions and classify traffic. Because of this, users MUST utilize real application traffic for determining benchmarking performance.

Due to the dynamic nature of the environment in which these devices are being deployed, this document will not explicitly state the application protocols or versions to be used for this methodology. While this is left to the discretion of the end user, there are several guidelines that SHOULD be used when determining the breadth and depth of application protocols to be used:

- The traffic generation pattern SHOULD contain all protocols that may be present in the final production deployment.
- The percentage of each protocol SHOULD approximate the percentage seen in the final production deployment.
- The application traffic SHOULD be unique traffic flows and not simply ones and zeros.

There are numerous tools available providing detailed information on the traffic flows in networks. A description or definition of these tools is outside the scope of this document.

3.4. Multiple Client/Server Testing

In actual network deployments, connections are being established between multiple clients and multiple servers simultaneously. RFC 3511 [3] specifies that connections must be initiated in a round-robin fashion, but in order to replicate performance in live networks, this method SHOULD NOT be used. The connection sequence ordering scenarios a device will see on a live network will likely be much less deterministic. Thus, users SHOULD setup the test equipment to issue requests at random to the virtual servers rather than in a predictable round-robin fashion. This method will help to appropriately reflect live network deployment behavior in the test setup.

3.5. Network Address Translation

Many content-aware devices are capable of performing Network Address Translation (NAT)[1]. If the final deployment of the DUT will have this functionality enabled, then the DUT MUST also have it enabled during the execution of this methodology. It MAY be beneficial to
perform the test series in both modes in order to determine the performance differential when using NAT. The test report MUST indicate whether NAT was enabled during the testing process.

3.6. TCP Stack Considerations

As with RFC 3511 [3], TCP options SHOULD remain constant across all devices under test in order to ensure truly comparable results. This document does not attempt to specify which TCP options should be used, but all devices tested SHOULD be subject to the same configuration options.

3.7. Other Considerations

Various content-aware devices will have widely varying feature sets. In the interest of realistic test results, the DUT features that will likely be enabled in the final deployment SHOULD be used. This methodology is not intended to advise on which features should be enabled, but to suggest using actual deployment configurations.

4. Benchmarking Tests

4.1. Maximum Application Connection Establishment Rate

4.1.1. Objective

To determine the maximum rate through which a device is able to establish application-specific sessions as defined by RFC 2647 [1].

4.1.2. Setup Parameters

The following parameters MUST be defined for all tests:

4.1.2.1. Transport-Layer Parameters

- Aging Time: The time, expressed in seconds that the DUT will keep a connection in its state table after receiving a TCP FIN or RST packet.

- Maximum Segment Size: The size in bytes of the largest segment which may be sent over a TCP connection.

4.1.2.2. Application-Layer Parameters

- Protocol List: A listing of the layer 4 through 7 protocols present in a given test run.
4.1.3. Procedure

The test SHOULD generate application network traffic that meets the conditions of Section 3.3. The traffic pattern SHOULD begin with an application session establishment rate of 5% of expected maximum. The test SHOULD be configured to increase the attempt rate in units of 5% up through 110% of expected maximum. The duration of each loading phase SHOULD be at least 30 seconds. This test MAY be repeated, each subsequent iteration beginning at 5% of expected maximum and increasing session establishment rate to 10% more than the maximum observed from the previous test run.

This procedure MAY be repeated any number of times with the results being averaged together.

4.1.4. Measurement

The following metrics MAY be determined from this test, and SHOULD be observed for each application protocol within the traffic mix:

4.1.4.1. Maximum Application Session Establishment Rate

The test tool SHOULD report the maximum rate at which application sessions were established.

4.1.4.2. Application Session Setup Time

The test tool SHOULD report the minimum, maximum and average application setup time.

4.1.4.3. Application Session Response Time

The test tool SHOULD report the minimum, maximum, and average application session response times.

4.1.4.4. Application Session Time To Close

The test tool SHOULD report the minimum, maximum, and average application session time to close.

4.1.4.5. Application Latency

The test tool SHOULD report the minimum, maximum and average amount of time an application packet takes to traverse the DUT.
4.2. Application Throughput

4.2.1. Objective

To determine the maximum rate through which a device is able to forward packets when using realistic and stateful applications.

4.2.2. Setup Parameters

The following parameters MUST be defined and reported for all tests:

4.2.2.1. Parameters

The same transport and application parameters as described in Section 4.1.2 MUST be used.

4.2.3. Procedure

This test will attempt to send application data through the device at a session rate of 30% of the maximum established as observed in Section 4.1. This procedure MAY be repeated with the results from each iteration averaged together.

4.2.4. Measurement

The following metrics MAY be determined from this test, and SHOULD be observed for each application protocol within the traffic mix:

4.2.4.1. Maximum Throughput

The test tool SHOULD report the minimum, maximum and average application throughput.

4.2.4.2. Packet Loss

The test tool SHOULD report the number of network packets lost or dropped from source to destination.

4.2.4.3. Application Setup Time

The test tool SHOULD report the minimum, maximum, and average amount of time necessary before an application may begin transmitting data.

4.2.4.4. Application Response Time

The test tool SHOULD report the minimum, maximum, and average response time of the application session.
4.2.4.5. Application Session Time To Close

The test tool SHOULD report the minimum, maximum and average amount of time for an application session to fully close.

4.2.4.6. Application Latency

The test tool SHOULD report the minimum, maximum and average amount of time an application packet takes to traverse the DUT.

4.3. Denial of Service Attack Mitigation

4.3.1. Objective

To determine the effects of a TCP SYN Flood Denial-of-Service (DoS) attack on application session performance.

4.3.2. Setup Parameters

The same parameters must be used for Transport-Layer and Application Layer Parameters previously specified in Section 4.1.2 and Section 4.2.2, respectively. Additionally, the following parameters MUST be defined and reported for all tests:

- SYN attack rate: Rate, expressed in packets per second at which the DUT will receive TCP SYN packets.[3]

4.3.3. Procedure

This test will utilize the procedures specified previously in Section 4.1.3 and Section 4.2.3. When performing the procedures listed previously, during the steady-state time, the test should generate TCP SYN packets at the rate defined by the SYN attack rate parameter described above. The test tool MUST NOT respond to the TCP SYN packets with TCP SYN/ACK packets. This procedure SHOULD be performed with the TCP SYN packets originating from a single host, as well as from multiple hosts.

Both procedures SHOULD be run with and without the feature enabled on the DUT to determine the affects of the DoS attack on the baseline metrics previously derived. Additionally, the test MAY be configured to generate other denial of service attacks, including distributed. This document does not attempt to specify which additional scenarios should be tested.
4.3.4. Measurement

For each protocol present in the traffic mix, in addition to the metrics specified by Section 4.1.4 and Section 4.2.4, the following metrics MAY be determined from this test:

4.3.4.1. False Positives

Record this measurement of the number of application sessions that were failed due because of false detection as denial of service attack.

4.3.4.2. False Negatives

Record the number of TCP SYN packets as part of the DoS stream that were allowed to pass through the DUT.

4.4. Malicious Traffic Mitigation

4.4.1. Objective

To determine the effects on performance that malicious traffic may have on the DUT.

4.4.2. Setup Parameters

The same parameters must be used for Transport-Layer and Application Layer Parameters previously specified in Section 4.1.2 and Section 4.2.2, respectively. Additionally, the following parameters MUST be defined and reported for all tests:

- Attack List: A listing of the malicious traffic that was generated by the test.

4.4.3. Procedure

This test will utilize the procedures specified previously in Section 4.1.3 and Section 4.2.3. When performing the procedures listed previously, during the steady-state time, the tester should generate malicious traffic representative of the final network deployment. The mix of attacks MAY include software vulnerability exploits, network worms, back-door access attempts, network probes and other malicious traffic.

If a DUT may be run with and without the attack mitigation, both procedures SHOULD be run with and without the feature enabled on the DUT to determine the affects of the malicious traffic on the baseline metrics previously derived. If a DUT does not have active attack
mitigation capabilities, this procedure SHOULD be run regardless. Certain malicious traffic could affect device performance even if the DUT does not actively inspect packet data for malicious traffic.

4.4.4. Measurement

For each protocol present in the traffic mix, in addition to the metrics specified by Section 4.1.4 and Section 4.2.4, the following metrics MAY be determined from this test:

4.4.4.1. False Positives

Record this measurement of the number of application transactions that were failed due to false detection as malicious traffic. This measurement has little meaning for DUTs that do not actively block malicious traffic.

4.4.4.2. False Negatives

This measurement is the number of malicious attacks that were passed through the DUT. This measurement has little meaning for DUTs that do not actively block malicious traffic.

4.5. Malformed Traffic Mitigation

4.5.1. Objective

To determine the effects on performance and stability that malformed traffic may have on the DUT.

4.5.2. Setup Parameters

The same parameters must be used for Transport-Layer and Application Layer Parameters previously specified in Section 4.1.2 and Section 4.2.2.

4.5.3. Procedure

This test will utilize the procedures specified previously in Section 4.1.3 and Section 4.2.3. When performing the procedures listed previously, during the steady-state time, the tester should generate malformed traffic at all protocol layers. This is commonly known as fuzzed traffic. This test SHOULD be run on a DUT regardless of whether it has built-in mitigation capabilities.
4.5.4. Measurement

For each protocol present in the traffic mix, the metrics specified by Section 4.1.4 and Section 4.2.4 MAY be determined. This data may be used to ascertain the effects of fuzzed traffic on the DUT.

5. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see the update of RFC 2434 [6] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

6. Security Considerations

The purpose of this document is to provide a methodology for benchmarking content-aware network interconnect devices. While this document does suggest running some tests utilizing software vulnerability exploits and network attacks, the primary purpose is to determine the effects on performance rather than assess the security of the DUTs themselves. Thus, security is outside the scope of this document.

7. References

7.1. Normative References


7.2. Informative References


Author’s Address

Mike Hamilton
BreakingPoint Systems
Austin, TX 78717
US

Phone: +1 512 636 2303
Email: mhamilton@breakingpoint.com