Pulse Secure®

WHITE PAPER

Creating Simple TrafficScript Rules with the Pulse Secure Virtual Traffic Manager

What Is TrafficScript?

TrafficScript is the customization language in Pulse Secure Virtual Traffic Manager (vTM). It makes it easy to create powerful traffic management rules that define how Pulse Secure vTM manages application traffic.

For example, TrafficScript can be configured with Pulse Secure vTM to:

- Recognize different types of requests and load-balance them to different servers
- Inspect outgoing responses and rewrite them if necessary
- Apply different Pulse Secure vTM features (e.g., bandwidth management, session persistence, etc.) to different requests or types of traffic

In essence, TrafficScript transforms load-balancing solutions into a much more flexible toolkit for inspecting and modifying traffic in real time. TrafficScript can specify exactly how each transaction to is handled; it's a little like being able to configure the traffic manager especially for each transaction it manages.

TrafficScript is quick and easy for network and application staff to use, so it is often used to rapidly fix complex problems such as security vulnerabilities, application bugs, or application incompatibilities.

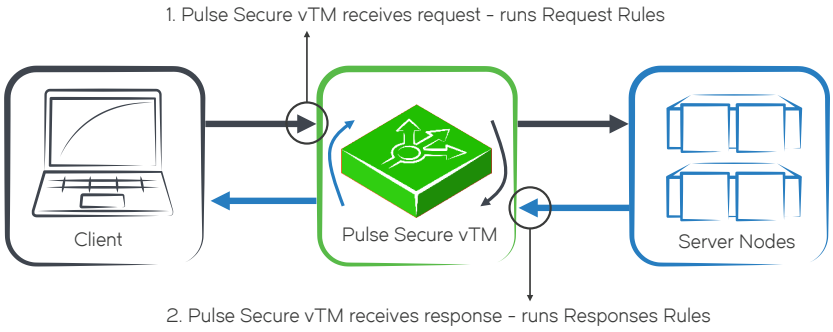

Figure 1. Example of how Pulse Secure Virtual Traffic Manager uses rules before and after load-balancing traffic to server nodes.

How Does TrafficScript Work?

TrafficScript is a programming language used to create rules. A TrafficScript rule can inspect and modify traffic, and activate and control Pulse Secure vTM. (See Figure 1.) It can also:

- Inspect all aspects of a request or response
- Modify any aspect of a request or response
- Activate (or deactivate) features of Pulse Secure vTM specifically for that request or response
- Control how Pulse Secure vTM handles the transaction, sending it to a particular set of servers, answering directly, or retrying the transaction if an error is detected

```
$ip = request.getRemoteIP();

# Reverse the IP, and append ".sbl-xbl.spamhaus.org".
$bytes = string.dottedToBytes( $ip );
$bytes = string.reverse( $bytes );
$query = string.bytesToDotted( $bytes ).".sbl-xbl.spamhaus.org";

if( net.dns.resolveHost( $query ) ) {
  log.warn( "Connection from IP ".$ip.": known Spam; dropping" );
  connection.sleep( 10 );
  connection.discard();
}
```

Figure 2: Example TrafficScript to detect and discard SMTP/Email traffic from known spam sources.

The TrafficScript language syntax is simple and familiar; it looks just like a simplified version of Perl, Java, C/C# etc. This makes it very easy to learn quickly.

It integrates closely with Pulse Secure Virtual Traffic Manager and there is a comprehensive set of helper functions to make it easy to process transaction data. The HTTP processing is particularly strong; Pulse Secure vTM manages HTTP complexities like keepalives, header parsing, compressed responses, and data chunking for the rule, so the TrafficScript administrator need not be concerned with them. TrafficScript also provides XML processing functions to manage complex XML data. (See Figure 2.)

TrafficScript is not just for HTTP (web) traffic. It can be used to inspect, manipulate, and manage any TCP or UDP protocol. There are helper functions to get and set raw data streams, and functions to inspect and process binary data as well as ASCII.

For example, organizations like Spamhaus publish lists of known spam and exploit sources using DNS. The following rule can be used to manage SMTP (email) traffic, checking each remote client against the Spamhaus black list[1]: If the connection comes from a known Spam source, this TrafficScript rule records a message, pauses the connection for 10 seconds, and then discards it. Pausing the connection prevents the remote mail server from continually retrying to connect—a technique known as a 'Tar Pit'.

What can be Controlled with TrafficScript?

TrafficScript can inspect each request and response, and then precisely define how the transaction is managed. TrafficScript can control:

- Which server(s) should process the request
- How the request or response should be rewritten (if desired)
- If the request should be retried if it fails
- What sort of session persistence should be used for each request
- How much bandwidth the request and response should use
- If the HTTP content cache should be used
- How many of each type of request should be admitted per second (or per minute)
- How performance should be monitored (using SLM)
- What information about the transaction should be logged
- Figure 3 shows examples of simple, yet powerful rules created with TrafficScript

Pulse Secure vTM creates adaptive rules that only take effect when traffic levels are high, servers are overloaded, or during particular times of the day. TrafficScript can be implemented in order to achieve the desired traffic management policy.

[1] www.spamhaus.org: To check IP 10.11.12.13, perform a DNS request for 13.12.11.10.sbl-xbl.spamhaus.org. A non-empty response indicates the IP is a known Spam source. For testing, refer to Latest Listings for known Spam IP addresses. See Spamhaus Datafeed Service for commercial or high-volume usage.

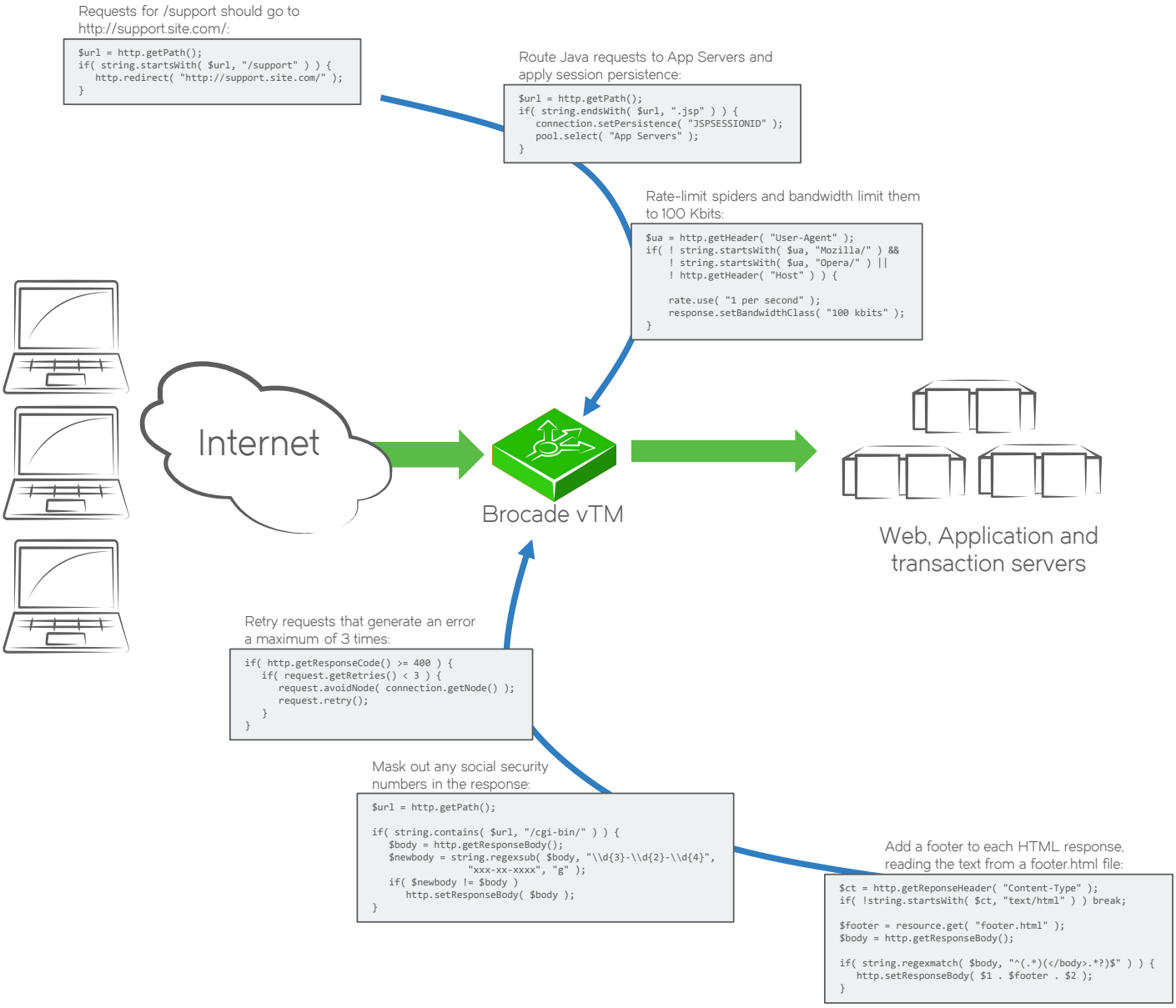

Figure 3: TrafficScript can be used to create powerful rules for content-based routing and application control.

TrafficScript is Easy to Use

Pulse Secure RuleBuilder makes it very easy to get started with TrafficScript; it's a powerful user interface tool that can be used to create a wide range of TrafficScript rules without needing any programming experience. (See Figure 4.)

After the first rules have been created by RuleBuilder, then they are connected into the TrafficScript language and processed to becoming a fully-fledged TrafficScript user.

Why not use a Programming Language like TCL?

An embedded language for traffic management application must be fast and extremely efficient when managing memory. Because traffic management rules are executed on every single connection:

- The traffic management rule must be compiled into an efficient, internal form rather then interpreted each time; run-time checks (e.g., types, function overloading, etc.) should be minimized

- Start-up time must be very low, because each rule is run thousands of times each second
- The language implementation must not copy memory unnecessarily, rather it should use reference counting for memory buffers to minimize the number of times request and response data is copied

General-purpose scripting languages have features like complex data types, objects, function overloading, and run-time evaluation. These add unnecessary overhead, reducing the language's performance and efficiency.

The design of TrafficScript is optimized for the task of managing network traffic data, and TrafficScript is very tightly integrated into Pulse Secure vTM. It uses Pulse Secure vTM to do all of the complex protocol handling, empowering the TrafficScript administrator to write simple rules that run efficiently.

Third-party Languages Require Complex, Event-driven Programming Models

Environments that use languages like TCL cannot do anything that might pause or wait, because that would block the entire Pulse Secure Virtual Traffic Manager. Each rule has to be broken into tiny parts and assigned to different events. This is a difficult, inefficient programming model, and makes simple tasks like reading an HTTP response or processing an XML stream extremely awkward.

The TrafficScript environment handles blocking operations seamlessly. For example, a rule is suspended when it asks for more data and is seamlessly restarted when the data is available.

Pulse Secure Virtual Traffic Manager has two simple events that can trigger rules (start of request, start of response), making it extremely easy and intuitive to write concise, simple TrafficScript rules.

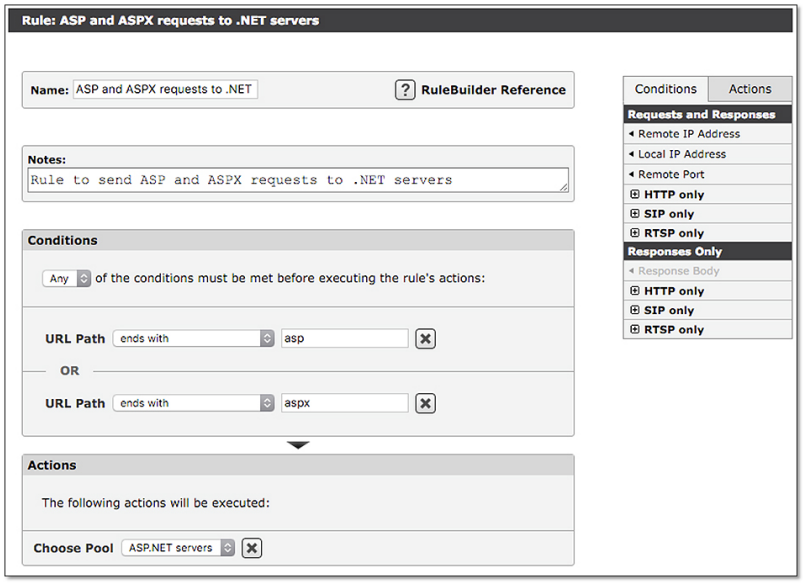

Figure 4: The Pulse Secure RuleBuilder is a graphic tool for creating traffic management rules.

About Pulse Secure

Pulse Secure networking solutions help organizations achieve their critical business initiatives as they transition to a world where applications and information reside anywhere. Today, Pulse Secure is extending its proven data center expertise across the entire network with open, virtual, and efficient solutions built for consolidation, virtualization, and cloud computing. Learn more at www.pulsesecure.net.

Corporate and Sales Headquarters
Pulse Secure LLC
2700 Zanker Rd. Suite 200
San Jose, CA 95134
www.pulsesecure.net

Pulse Secure®