ESA, SMA, and WSA Grep with Regex to Search Logs

Document ID: 117968

Contributed by Jeff Richmond, Cisco TAC Engineer.
Jul 16, 2014

Contents

Introduction

This document describes how to use regular expressions (regex) with the ***grep*** command in order to search logs.

Prerequisites

The information in this document is based on these software and hardware versions:

- Cisco Web Security Appliance (WSA)
- Cisco Email Security Appliance (ESA)
- Cisco Security Management Appliance (SMA)

Grep with Regex

Regex can be a powerful tool when used with the ***grep*** command to search through logs available on the appliance, such as Access Logs, Proxy Logs, and others. You can search the logs based on the website, or any part of the URL, and user names with the ***grep*** CLI command.

Here are some common scenarios where you can use regex with the ***grep*** command in order to assist with troubleshooting.

Scenario 1: Find a Particular Website in the Access Logs

The most common scenario is when you attempt to find requests that are made to a website in the access logs of the WSA.

Here is an example:

Connect to the appliance via Secure Shell (SSH). Once you have the prompt, enter the ***grep*** command in order to list the available logs.

```
CLI> grep
```

Enter the number of the log you wish to ***grep***.

```
[]> 1 (Choose the # for access logs here)
```

Enter the regular expression to ***grep***.

```
[]> website\.com
```

Scenario 2: Attempt to Find a Particular File Extension or Top−Level Domain

You can use the ***grep*** command in order to find a particular file extension (.doc, .pptx) in a URL or a top−level domain (.com, .org).

Here is an example:

In order to find all URLs that end with .crl, use this regex:

```
\.crl$
```

In order to find all URLs that contain the file extension .pptx, use this regex:

```
\.pptx
```

Scenario 3: Attempt to Find a Particular Block for a Website

When you search for a particular website, you might also search for a particular HTTP response.

Here is an example:

If you want to search for all TCP_DENIED/403 messages for domain.com, use this regex:

```
tcp_denied/403.*domain\.com
```

Scenario 4: Find a Machine Name in the Access Logs

When you use the NTLMSSP authentication scheme, you might encounter an instance where a User Agent (Microsoft NCSI is the most common) incorrectly sends machine credentials instead of user credentials when it authenticates. In order to track down the URL/User Agent that causes this issue, use regex with ***grep*** in order to isolate the request made when the authentication occurred.

If you do not have the machine name that was used, use ***grep*** and find all machine names that were used as user names when authenticating with this regex:

```
\$@
```

Once you have the line where this occurs, grep for the specific machine name that was used with this regex:

```
machinename\$
```

The first entry that appears should be the request that was made when the user authenticated with the machine name instead of the user name.

Scenario 5: Find a Specific Time Period in the Access Logs

By default, access log subscriptions do not include the field that shows the human readable date/time. If you want to check the access logs for a particular time period, complete these steps:

1. Look up the UNIX timestamp from a site such as Online Conversion.
2. Once you have the timestamp, search for a specific time within the Access Logs.

Here is an example:

A Unix timestamp of ***1325419200*** is equivalent to ***01/01/2012 12:00:00***.

You can use this regex entry in order to search the access logs close to 12:00 on January 1, 2012:

13254192

Scenario 6: Search for Critical or Warning Messages

You can search for critical or warning messages in any available logs, such as proxy logs or system logs, with regular expressions.

Here is an example:

In order to search for warning messages in the proxy logs, enter this regex:

```
CLI> grep
```

Enter the number of the log you wish to ***grep***.

```
[]> 17 (Choose the # for proxy logs here)
```

Enter the regular expression to ***grep***.

```
[]> warning
```

Updated: Jul 16, 2014 Document ID: 117968