Logfile Guide

Date: 2012-09-11 08:22 UTC

Amalienbadstrasse 42/Bau 52, 76227 Karlsruhe, Germany

http://www.sophos.com/

Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to <nsg-docu@sophos.com>.

Table of Contents

Chapter 1

Introduction

This document describes logs that are generated by applications running on Sophos UTM 9.

Logfile Format

The standard log format consists of two parts:

```
<Fixed Part> <Variable Part>
```

Fixed Part

The fixed part is always the same for each facility of the product. It's the same for logs created by the SMTP proxy, VPN, or user authentication, to name just a few.

```
<SYSLOGFIX> <EVENT_TYPE_ID> <SEVERITY_LEVEL> <SYSTEM> <SUBSYSTEM> <EVENT_TYPE_NAME>
```

The elements always appear in this order in the logline.

Each log consists of multiple key-value pairs that can easily be parsed, for example by means of the following code written in Perl:

```
# expects:  log line in astaro log format
# returns:  hash reference containing key-value pairs
sub ParseAstaroMessage {
  my $alfdata = $_[0];  # astaro log format data
  my $data;
  while ($alfdata =~ /(\S+)="(.*?)"/og) {
    $data->{$1} = $2;
  }
  return $data;
}
```

Note that some characters occuring in the *value* part are URL encoded, for example:

```
name="joe says %22hi%22"
```

The example above means that the parameter *name* has the value *joe says "hi"* (quotes included).

The following characters are URL encoded:

- Characters lesser than whitespace, i.e., No. 32 of the ASCII table.
- Characters greater than tilde (~), i.e., No. 126 of the ASCII table.
- Double quotes (")
- Percent (%)

Except for SYSLOGFIX, all elements in the *<Fixed Part>* consist of name/value pairs having the following syntax:

```
fixed_part = name_value_pair*
name_value_pair = name "=" value
```

```
name = [a-z]|[A-Z]|[0-9]+
value = '"' [a-z]|[A-Z]|[0-9][ ]+ '"'
```

SYSLOGFIX

Part of the log entry written automatically by Syslog. This includes

- the date and time of occurrence,
- the hostname,
- the application from which the event originated (by default the name of the binary),
- the facility used when logging.

Example:

```
yyyy:mm:dd-hh:mm:ss hostname process[PID]:
2006:11:5-15:22:08 ctsync-1 ulogd[1116]:
```

EVENT_TYPE_ID

Identifies the type of the event. This ID is a global unique ID which identifies the type of event and logline format across the UTM 9 logging framework. This ID is a 4 character string representing the SYSTEM, SUBSYSTEM and EVENT_ID parts of the log:

```
     +---- S = SYSTEM_ID - [0-9]{1,1}
     |
     |+--- U = SUBSYSTEM_ID - [0-9]|[A-Z]{1,1}
     ||
     ||+--- EE = EVENT_ID - [0-9]|[A-Z]{2,2}
     |||
     |||
id="SUEE"
```

For example the SYSTEM "SecureWeb" has the SYSTEM_ID "0". The SUBSYSTEM "pop3" has the SUBSYSTEM_ID "1" and the EVENT "Virus detected" has the EVENT_ID "08". Therefore, the EVENT_TYPE_ID of this logline has the value "0108":

```
sys="SecureWeb"(0) + sub="pop3"(1) + name="Virus detected"(08) => id="0108"
```

Note – The EVENT_ID "00" is reserved for debug messages which are not restricted to fulfill the name/value requirement in the *<Variable Part>*. For the *<Fixed part>* the name/value rule is also

valid for debug messages.

SEVERITY_LEVEL

The severity level (parameter `severity`) indicates how serious the event might influence the system and/or its users.

Important Note – Note that only logs of severity *info* are necessary for reporting purposes.

The following severity levels are available:

- *debug*—information is only interesting for Support/QA/EE and can be safely ignored by the customer (normally, these entries are not even written to the log unless a special option is enabled).
- *info*—informational messages which are created during normal system operation. These entries mostly include information about allowed traffic (mails, packets, etc.).
- *warn*—contains mostly information about blocked traffic. It requires administrator's attention but is still not an indication of a malfunction. Such entries could be very useful in detecting errors in the user's configuration.
- *error*—some unexpected condition has occurred but the system is of the opinion that it can recover from it and that the reported condition is temporary. Typical examples are non running virus scanners, unreachable Cobion servers etc.
- *crit*—a condition is detected which impairs the overall system stability and it is not clear if the system can recover without the user/support intervention. A nice example of such an issues is when a HD partition fills up.

SYSTEM

The UTM system (*sys*) which reported the event. Possible values include:

- *SecureWeb* (ID = 0)
- *SecureMail* (ID = 1)
- *SecureNet* (ID = 2)
- *System* (ID = 3)

This list is an enumerated type data type—no other values are allowed (as they are accepted by all involved instances).

SUBSYSTEM

Identifies the subsystem (*sub*) in the aforementioned system.
Possible values for the SecureWeb system include:

- *http* (ID = 00)
- *ftp* (ID = 01)

Possible values for the SecureMail system include:

- *smtp* (ID = 10)
- *pop3* (ID = 11)
- *mailmanager* (ID = 12)

Possible values for the SecureNet system include:

- *packetfilter* (ID = 20)
- *ips* (ID = 21)
- *vpn* (ID = 22)—this value has the parameter *variant*, which can take one of the following values:
  - *ipsec*
  - *pptp*
  - *l2tp*
  - *ssl*

Possible values for the SYSTEM system include:

- *auth* (ID = 30)
- *confd* (ID = 31)
- *CFF* (ID = 33)
- *av* (ID = 34)
- *certmanager* (ID = 35)
- *cobion* (ID = 36)
- *up2date* (ID = 37)
- *ha* (ID = 38)
- *accounting* (ID = 39)
- *service_monitor* (ID = 40)
- *wireless security* (ID = 41)

The SUBSYSTEM is also an enum, depending on the SYSTEM.

EVENT_TYPE_NAME

Identifies the type of the event (*name*) in a human readable form. The name of the event type corresponds with the EVENT_TYPE_ID

The name of the event describes this event ID in human readable form. Some examples:

- *web request blocked, virus detected* (ID = 0056)
- *file blocked, virus detected* (ID = 0104)
- *email delivered* (ID = 1004)

Variable Part

The variable part has the same syntax (name/value pairs) for each facility but differs in data content.
Format of the variable part is:

```
variable_part = name_value_pair*
name_value_pair = name "=" value
name = [a-z]|[0-9]+
value = '"' [http://www.rfc-editor.org/rfc/rfc1738.txt url_escaped_octet]* '"'
```

Each event type has his own set of values which must appear in the log lines describing it. The ordering of the entries is considered significant (to simplify parsing).
Reserved names for the variable log file part:
Actions:

- **action**: what action happens whith the content. (e.g. *drop*, *skip*, *blocked*)
- **reason**: the reason why an action occoured (e.g. *DENIED*)
- **caller**: name of the instance that has called a module or a functionality (e.g. *webadmin*)
- **engine**: engine that did the action (e.g. *local*, *radius*)

Sizes/Numbers:

- **seq**: continuous and unique number generated by the application (e.g. *1224323224*)
- **length**: length of data (e.g. *40*, *35MB*)
- **count**: number of occurrence (e.g. *12*)

Network:

- **srcip**: source ip adress (e.g. *86.16.111.5*)
- **dstip**: destination ip adress (e.g. *85.216.17.180*)
- **proto**: protocol number as defined by IANA (see http://www.iana.org/assignments/protocol-numbers). For example, *6* for TCP)
- **srcport**: source port (e.g. *16972*)
- **dstport**: destination port (e.g. *24201*)
- **initf**: incoming interface (e.g. *eth2*)
- **outitf**: outgoing interface (e.g. *eth1*)
- **srcmac**: source MAC adress (e.g. *00:04:76:10:a7:04*)
- **dstmac**: destination MAC adress (e.g. *00:01:5c:23:4d:02*)
- **serverip**: ip adress of a server (e.g. *192.168.17.22*)
- **server**: name of a server (e.g. *pop3.example.com*)

Netfilter:

- **tcpflags**: (e.g. *RST*)
- **tcpseq**: (e.g. *3908017709*)
- **tcpack**: (e.g. *3066534274*)
- **tcpwindow**: (e.g. *65467*)

Web:

- **srcip**: sender IP address
- **user**: username
- **url**: URL
- **size**: size of file
- **method**: GET or POST
- **engine**: scanner name (1: Open Source AV engine, 2: Astaro AV engine, 3: Hardware accelerated AV engine).
- **virus**: name of the virus caught
- **extension**: file extension that is responsible for the request being blocked
- **filename**: name of the file that is responsible for the request being blocked
- **category**: Astaro Sub-Category ID
- **categoryname**: category name as given in WebAdmin
- **entry**: entry in the content filter blacklist if everything is blocked and only whitelist is used.

E-Mail:

- **subject**: the subject of a message (e.g. *Here is a loveletter...*)
- **from**: from whom is the message (e.g. *somebody@somewhere*)
- **to**: the recipient of the message (e.g. *jdoe@example.com*)
- **messageid**: message id (e.g. *12345*)
- **virus**: name of a virus (e.g. *EICAR-Signature-File*)

Users:

- **accountid**: id of an account (e.g. *123452*)
- **userid**: id of a user (e.g. *3434322*)
- **user**: name of a user (e.g. *admin*)

Accounting:

- **connstart**: begin of the connection being accounted (Unix time, that is, the number of seconds elapsed since January 1, 1970)
- **connend**: end of the connection being accounted (Unix time, that is, the number of seconds elapsed since January 1, 1970)
- **trafficin**: incoming aggregated traffic in bytes
- **trafficout**: outgoing aggregated traffic in bytes

Chapter 2

Authentication Service (AUA)

Daemon Started Successfully

Description: This log entry is being generated when the Astaro User Authentication has started successfully.

ID: 3001

```
2006:12:13-12:00:51 (none) aua[2988]:  id="3001" severity="info"
sys="System" sub="auth" name="Daemon started successfully"
```

Authentication Successful

ID: 3004

Description: This log entry is being generated when a component has successfully authenticated a user at the authentication daemon.

- `srcip`: source IP address of authentication client (e.g. user workstation)
- `user`: username used for authentication
- `caller`: internal module that requested the authentication (webadmin, portal, openvpn, pptp, l2tp, socks, http, smtp, system)
- `engine`: authentication method which successfully authenticated the user (local, radius, tacacs, edirectory, adirectory, ldap)

```
2006:12:13-12:15:26 (none) aua[5297]:  id="3004" severity="info"
sys="System" sub="auth" name="Authentification successful"
srcip="192.168.2.96" user="admin" caller="webadmin" engine="local"
```

Authentication Failed

ID: 3005

Description: This log entry is being generated when a component failed to authenticate a user at the authentication daemon.

- `srcip`: source IP address of authentication client (e.g. user workstation)
- `user`: username used for authentication
- `caller`: internal module that requested the authentication (webadmin, portal, openvpn, pptp, l2tp, socks, http, smtp, system)
- `reason`: text giving a reason that (and possibly why) the user was denied

```
2006:12:15-10:40:46 (none) aua[5164]:  id="3005" severity="warn"
sys="System" sub="auth" name="Authentification failed"
srcip="192.168.2.96" user="admin" caller="webadmin" reason="DENIED"
```

Informational Message

ID: 3006

Description: This log entry is being generated when a general info message is issued by the authentication daemon.

- `event`: informational message text

```
2006:12:15-10:41:14 (none) aua[5175]:  id="3006" severity="info"
sys="System" sub="auth" name="do_auth:  authentication succeeded
with method local, checking authorization now"
```

Debug Message

ID: 3007

Description: This log entry is being generated when a general debug message is issued by the authentication daemon.

- `event`: Debug message text

```
2006:12:15-11:00:57 (none) aua[5350]:  id="3007" severity="debug"
sys="System" sub="auth" name="Master:  waiting for new connection."
```

Warning: file_get_contents(http://wiki.intranet.astaro.de/index.php?title=Template:subst&action=raw) [function.file-get-contents]: failed to open stream: HTTP request failed! HTTP/1.0 404 Not Found

```
in /var/www/getpage.php on line 38
```

Chapter 3

FTP

Fixed fields

- `srcip`: client IP address
- `dstip`: server IP address
- `url`: full url ftp://host/path/file
- `user`: username
- `size`: size of file

Optional fields:

- `extension`: forbidden file extension
- `virus`: virusname

IDs:

- 0101 send file, not scanned
- 0102 send file, clean
- 0103 error
- 0104 file blocked, virus detected
- 0105 file blocked, forbidden extension

Example:

```
 frox[10685]:  id="0104" severity="info" sys="SecureWeb" sub="ftp" name="file
blocked, virus detected"
 srcip="192.168.12.34" dstip="10.8.16.108" url="ftp://10.8.16.108/pub/xyz.exe"
 user="anonymous" virus="W32/xyz"
```

Chapter 4

High Availability (HA) and Clustering

In the following we provide a comprehensive overview of all log events of the HA/Clustering system.

Switching to Master Mode

ID: 38B0

Description: This log entry is being generated when the system switches to master mode

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B0" severity="info"
sys="System" sub="ha" name="Switching to Master mode"
```

Switching to Slave Mode

ID: 38B1

Description: This log entry is being generated when the system switches to slave mode

```
2006:09:26-17:12:54 (none) ha_daemon[2137]:  id="38B1" severity="info"
sys="System" sub="ha" name="Switching to Slave mode"
```

Switching to Worker Mode

ID: 38B2

Description: This log entry is being generated when the system switches to worker mode

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B2" severity="info"
sys="System" sub="ha" name="Switching to Worker mode"
```

Grateful Take-over

ID: 38B4

Description: This log entry is being generated when there is a graceful take-over (old master retired, for example because of linkbeat failure, manual take-over or reboot)

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B4" severity="info"
sys="System" sub="ha" name="No active Master found,
initiating graceful takeover!"
```

Master Is Dead, Take-over

ID: 38B5

Description: This log entry is being generated when a slave detects that the master is dead and takes over

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B5" severity="info"
sys="System" sub="ha" name="Master is dead, taking over!"
```

Graceful Take-over Started

ID: 38B6

Description: This log entry is being generated when the master passes control to a slave, because it has better connectivity

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B6" severity="info"
sys="System" sub="ha" name="Slave 1 with better connectivity around,
initiating graceful takeover!"
```

Cluster: Slave Is Dead, Taking Over

ID: 38B7

Description: This log entry is being generated when a worker node goes into slave mode (on cluster)

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B7" severity="info"
sys="System" sub="ha" name="Slave is dead, taking over!"
```

Cluster: Link Failed on Slave, Graceful Take-over Started

ID: 38B8

Description: This log entry is being generated when the master passes the master role to a worker, because the worker has a better connectivity

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B8" severity="info"
sys="System" sub="ha" name="Worker 1 with better connectivity around,
initiating graceful takeover!"
```

HA Graceful Take-over After Up2Date

ID: 38B9

Description: This log entry is being generated when an up2date in HA was successful, and a takeover is triggered

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38B9" severity="info"
sys="System" sub="ha" name="HA up2date successful, initiating graceful
takeover"
```

Cluster Graceful Take-over After Up2Date

ID: 38Ba

Description: This log entry is being generated when an up2date in the cluster was successful, and a takeover is triggered

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38Ba" severity="info"
sys="System" sub="ha" name="Cluster up2date successful, initiating graceful
takeover"
```

Master-Master Condition, Comparing Time Running as Master

ID: 38Bb

Description: This log entry is being generated when this node goes into slave mode in favor to another master, because of a longer uptime

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38Bb" severity="info"
sys="System" sub="ha" name="Going slave mode in favour of node 1 (-10 sec)"
```

Master-Master Condition, Same Time Running as Master, Comparing Node ID

ID: 38Bc

Description: This log entry is being generated when this node goes into slave mode in favor to another master, because of a higher node ID

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38Bc" severity="info"
sys="System" sub="ha" name="Going slave mode in favour of node 1
(higher node id)"
```

Slave-Slave Condition, Comparing Time Running as Slave

ID: 38Bd

Description: This log entry is being generated when this node goes into worker mode in favor to another node, because of a longer uptime

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38Bd" severity="info"
sys="System" sub="ha" name="Going worker mode in favour of node 1 (-10 sec)"
```

Slave-Slave Condition, Same Time Running as Slave, Comparing Node ID

ID: 38Be

Description: This log entry is being generated when this node goes into worker mode in favor to another node, because of a higher node ID

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38Be" severity="info"
sys="System" sub="ha" name="Going worker mode in favour of node 1
(higher node id)"
```

Got User Take-over Request

ID: 38Bf

Description: This log entry is being generated when a takeover is triggered by the admin

```
2006:09:26-17:18:04 (none) ha_daemon[2137]:  id="38Bf" severity="info"
sys="System" sub="ha" name="Got user takeover request, initiating graceful
takeover!"
```

Node Is Alive

ID: 38C0

Description: This log entry is being generated when another node is detected on the net

```
2006:09:26-17:15:47 (none) ha_daemon[27289]:  id="38C0" severity="info"
sys="System" sub="ha" name="Node 1 is alive!"
```

Node Is Dead

ID: 38C1

Description: This log entry is being generated when a node is not seen anymore on the net

```
2006:09:26-17:14:34 (none) ha_daemon[27289]:  id="38C1" severity="info"
sys="System" sub="ha" name="Node 1 is dead, received no heart beats!"
```

Preempt Slave

ID: 38C2

Description: This log entry is being generated when a graceful takeover is triggered by preempt Slave.

```
2006:09:26-17:14:34 (none) ha_daemon[27289]:  id="38C2" severity="info"
sys="System" sub="ha" name="Preempt Slave 1, initiating graceful takeovers!"
```

Preempt Worker

ID: 38C3

Description: This log entry is being generated when a graceful takeover is triggered by preempt Worker.

```
2006:09:26-17:14:34 (none) ha_daemon[27289]:  id="38C3" severity="info"
sys="System" sub="ha" name="Preempt Worker 1, initiating graceful takeover!"
```

Web Request Delivered to Sender

Description: This log is being generated when a web request was successfully delivered.

ID: 0001

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET, POST, and other request methods
- `action`: pass
- `statuscode`: Statuscode delivered to the client
- `cached`: 1 if the content was delivered from the local cache
- `profile`: the profile assigned to the request, selected by source ip
- `filteraction`: Filteraction used for Virus/Content Scanning.
- `size`: Number of octets delivered to the client (without headers, so only body bytes are counted)
- `request`: memory address of the data structure used for this request. This address is added to almost all debugging messages from the proxy, so it is something like a *unique identifier*'

- `error`: In case something goes wrong during processing, this field may contain additional information what failed
- `category`: If URL blocking by URL is enabled in the filteraction, this field will contain all category numbers seperated by comma
- `categoryname`: As above, but with plain names seperated by comma instead of numbers

```
  2009:01:12-15:11:10 (none) httpproxy[5757]:  id="0001" severity="info" sys="SecureWeb"
sub="http" name="http access" action="pass" method="GET" srcip="10.128.129.192"
user="" statuscode="200" cached="0" profile="profile_1" filteraction="action_REF_DefaultHTTPCFFActio
size="9511"  request="0x9c6512a8" url="http://ads2.net2day.de/adjs.php?n=597887660&clientid=59"
error="" category="116,145,154,159,164,177" categoryname="Games,Search Engines,Web
Ads ,Forum/Bulletin Boards,Visual Search Engine,Content Server" content-type="application/x-
javascript"
```

Web Request Blocked due to Error

Description: This log is being generated when a web request was blocked due to an error.
ID: 0002

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST

```
httpproxy[...]:  id="0002" severity="info" sys="SecureWeb" sub="http"
srcip="192.168.2.104" user="user@group" url="http://www.astaro.com/"
size="2342" method="GET" name="web request could not be delivered due
to an error"
```

Web Request Blocked due to Virus

Description: This log is being generated when a web request was blocked due to a virus.
ID: 0056

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST
- `engine`: scanner name (ClamAV, Astaro AV).
- `virus`: name of the virus caught

- `action`: block
- `reason`: reason of the blocking

```
  2007:03:26-09:41:22 198.19.250.2 httpproxy[29200]:  id="0056" severity="info"
sys="SecureWeb" sub="http" name="web request blocked, virus detected"
  engine="2 " virus="Eicar" action="block" reason="infected" srcip="10.128.129.11"
user="" url="http://www.eicar.org/download/eicar_com.zip"
  method="GET" size="593"
```

Web Request Blocked due to URL Filter

Description: This log is being generated when a web request was blocked because a forbidden URL category was detected.
ID: 0060

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST
- `category`: Astaro Sub-Category ID
- `categoryname`: category name as given in WebAdmin
- `reputation`: either one of malicious, suspicious, unverified, neutral or trusted
- `action`: block
- `reason`: reason of the blocking

```
  2007:03:26-09:51:10 (none) httpproxy[3141]:  id="0060" severity="info" sys="SecureWeb"
sub="http"
  name="web request blocked, forbidden category detected" category="220" category-
name="Political Extreme / Hate / Discrimination"
  action="block" reason="category" srcip="10.128.129.12" user="" url="http://kkk.com/"
method="GET" size="0"
```

Web Request Blocked due to Reputation Limit

ID: 0061

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST
- `category`: Astaro Sub-Category ID

- `categoryname`: category name as given in WebAdmin
- `reputation`: either one of malicious, suspicious, unverified, neutral or trusted
- `action`: block
- `reason`: reason of the blocking

```
  2007:03:26-09:51:10 (none) httpproxy[3141]:  id="0060" severity="info" sys="SecureWeb"
sub="http"
  name="web request blocked, forbidden category detected" category="220" category-
name="Political Extreme / Hate / Discrimination"
  action="block" reason="category" reputuation="suspicious" srcip="10.128.129.12"
user="" url="http://kkk.com/" method="GET" size="0"
```

Web Request Blocked due to Blacklist

Description: This log is being generated when a web request was blocked due to a blacklisted URL.
ID: 0062

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST
- `entry`: entry in the content filter blacklist if everything is blocked and only whitelist is used.
- `action`: block
- `reason`: reason of the blocking

```
  2007:03:26-09:50:48 (none) httpproxy[3141]:  id="0062" severity="info" sys="SecureWeb"
sub="http" name="web request blocked, forbidden url detected"
  entry="heise.de" action="block" reason="blacklist" srcip="10.128.129.12" user=""
url="http://www.heise.de/" method="GET" size="0"
```

Web Request Blocked due to File Extension

Description: This log is being generated when a web request was blocked due to a forbidden file extension.
ID: 0064

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST

- `extension`: file extension that is responsible for the request being blocked
- `filename`: name of the file that is responsible for the request being blocked
- `action`: block
- `reason`: reason of the blocking

```
  2007:03:26-09:45:32 (none) httpproxy[3141]:  id="0064" severity="info" sys="SecureWeb"
sub="http"
  name="web request blocked, forbidden file extension detected" extension="exe"
filename="setup.exe" action="block"
  reason="extension" srcip="10.128.129.12" user="anyone"
  url="http://mozilla2.mirrors.tds.net/pub/mozilla.org/firefox/releases/2.0.0.3/win32/ar/setup.exe"
  method="GET" size="0"
```

Web Request Blocked due forbidden Mime Type

Description: This log is being generated when a web request was blocked due to a forbidden mime type

ID: 0065

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file
- `method`: GET or POST
- `action`: block
- `reason`: reason of the blocking
- `content-type`: forbidden mime type

```
  2007:03:26-09:45:32 (none) httpproxy[3141]:  id="0065" severity="info" sys="SecureWeb"
sub="http"
  name="web request blocked, forbidden mimetype detected" content-type="application/octet-
stream" action="block"
  srcip="10.128.129.12" user="anyone"  url="http://mozilla2.mir
```

Web Request Blocked due forbidden application

Description: This log is being generated when a web request was blocked due to a forbidden mime type

ID: 0066

- `srcip`: sender IP address
- `user`: username
- `url`: URL
- `size`: size of file

- `method`: GET or POST
- `action`: block
- `reason`: reason of the blocking
- `application`: forbidden application

```
  2007:03:26-09:45:32 (none) httpproxy[3141]:  id="0066" severity="info" sys="SecureWeb"
sub="http"
  name="web request blocked, forbidden application detected" content-type="application/octet-
stream" action="block"
  srcip="10.128.129.12" user="anyone"  application="facebook"
```

Common Log Fields

- `ID`: value is always *2101*
- `action`: the action being done on the packet:
  - possible value: *alert*|*drop*|*reject*
  - *alert* (alert only),
  - *drop* (packet dropped) or
  - *reject* (packet dropped, icmp "port unreachable" or TCP-RST was sent to original sender)
- `reason`: intrusion protection rule message text
- `group`: intrusion protection category from WebAdmin
- `srcip`: source IPv4 or IPv6 address of packet that triggered the alert
- `dstip`: destination IPv4 or IPv6 address of packet that triggered the alert
- `proto`: IP protocol of packet that triggered the alert
- `srcport`: source port of packet that triggered the alert
- `dstport`: destination port of packet that triggered the alert
- `sid`: Snort ID of alerting rule
- `class`: classification of rule
- `generator`: internal Snort module ID of module that issued the alert
- `msgid`: internal Snort index of alert message

Example:

```
2006:12:14-18:56:27 (none) snort[4138]:  id="2101" severity="warn"
sys="SecureNet" sub="ips" name="Intrusion protection alert" action="drop"
reason="BAD-TRAFFIC tcp port 0 traffic" group="410" srcip="192.168.2.95"
dstip="192.168.2.99" proto="6" srcport="46970" dstport="0" sid="524"
class="Misc activity" priority="3" generator="1" msgid="0"
```

Warning: file_get_contents(http://wiki.intranet.astaro.de/index.php?title=Template:subst &action=raw) [function.file-get-contents]: failed to open stream: HTTP request failed! HTTP/1.0 404 Not Found

```
in /var/www/getpage.php on line 38
```

Chapter 5

POP3

Notice: If an e-mail has more than one recipient, only the first recipient will be written to the *to* parameter. The *srcip* parameter contains the last ip address found in the received headers. Private networks and localhost are ignored. `0.0.0.0` is used, if no ip address was found.

There will be exactly one log line for each mail while it is being scanned.

Fixed Fields

- `from`: sender e-mail address
- `to`: first recipient e-mail address
- `subject`: subject
- `size`: size provided by server
- `srcip`: sender IP address
- `dstip`: IP address of pop3-server
- `uid`: uid provided by server

Optional Fields

- `ident`: present if the message got cached on the POP3 proxy. Contains the proxy-identification, which is built from the cluster node id and the spoolfile identifier: $node_id/$ident.
- `reason`: present if the message got quarantined. One of the following values:
  - `av`: AntiVirus
  - `as`: AntiSpam
  - `ext`: File Extension
  - `exp`: Expression Filter
  - `unscannable`: Attachment exceeds size limits or is encrypted
  - `sender_blacklist`: sender address is globally blacklisted or blacklisted by recipient
- `extra`: only if reason is present, contains an additional explanation of the quarantine reason.

IDs/Names

- 1100: email passed
- 1101: email quarantined

Example:

```
 pop3proxy[10685]:  id="1100" severity="info" sys="SecureMail" sub="pop3" name="email
passed"
 from="master@lab.hoteurope.de" to="user2@lab.hoteurope.de" subject="i am an
email" size="3559"
 srcip="1.2.3.4" dstip="10.8.16.108" uid="1193152748.33" ident="0/10685-1-1201528597"
```

Chapter 6

SMTP

General Format

```
id="[ID]"
severity="info"
sys="SecureMail"
sub="smtp"
name="[NAME]"
srcip="[SRCIP]"
from="[FROM]"
to="[TO]"
subject="[SUBJECT]"
queueid="[QUEUEID]"
size="[SIZE]"
(reason="[REASON]")
(extra="[EXTRA]")
```

All field data is in UTF-8 charset. Field data is URL-quoted (%XX). The '%' and '"' characters MUST be quoted, other bytes(!) CAN be quoted. So UTF-8 characters may appear as a sequence of coded byte values.

IDs/Names

Email Passed

ID: 1000

Name: email passed

Email Quarantined

ID: 1001

Name: email quarantined

Email Blackholed

ID: 1002

Name: email blackholed

Email Rejected

ID: 1003

Name: email rejected

Detailed Description

The placeholders in square brackets can have the following values:

- `srcip`: IPv4 message origin address as derived from the SMTP TCP socket
- `from`: envelope sender address, empty for bounces
- `to`: recipient address
- `subject`: The decoded message subject in UTF8 charset, whole item omitted if not available.
- `queueid`: The message ID in Exim format (xxxxxx-xxxxxx-xx, where x is of A-Za-z0-9). The whole item is omitted if the event is a pre-data reject.
- `size`: Message size in bytes, zero ("0") or "-1" if not available. Should be ignored in statistical calculations in the latter case.
- `reason`: Whole item omitted for "passed" mail, or one of those:
  - `av`: Malware
  - `as`: Spam Engine (currently Commtouch)
  - `ext`: Extension
  - `exp`: Expression
  - `mime`: MIME-type
  - `host_blacklist`: Host Blacklist
  - `sender_blacklist`: Sender Blacklist
  - `rdns_helo`: RDNS/HELO checks
  - `rbl`: RBL
  - `batv`: BATV
  - `address_verification`: Address verification
  - `spf`: SPF
  - `unscannable`: Message contains encrypted archive
  - `other`: something else
- `extra`: Whole item omitted for "passed" mail, or misc text that contains extended info, e.g. malware name, file extension, MIME-type etc.

Chapter 7

VPN

These events can be found in the following logfiles:

- `ipsec.log`
- `pptp.log`
- `openvpn.log`
- `html5vpn.log`

Connection Started

Description: This log entry is being generated when a user successfully connects to a remote access service

ID: 2201

- `username`: username of the user that logged in
- `variant`: VPN variant (ipsec, pptp, l2tp, ssl, clv)
- `srcip`: public source IP address from which the client connects
- `virtual_ip`: private pool IP address assigned to the user during the session

Only `variant=clv`:

- `service`: "HTML5 VPN"
- `type`: VNCRDesktop | VNCWebapp | PTYSSH | PTYTelnet | VNC
- `sessionid`: Session identifier
- `sessionname`: Name of the connection as defined in Webadmin

```
2009:09:15-10:54:07 utm pppd-pptp[5467]:  id="2201" severity="info"
sys="SecureNet" sub="vpn" event="Connection started" username="bertram"
variant="pptp" srcip="192.168.2.96" virtual_ip="10.242.1.2"
```

Connection Terminated

Description: This log entry is being generated when a user disconnects from a remote access service

ID: 2202

- `username`: username of user that logged in
- `variant`: VPN variant (ipsec, pptp, l2tp, ssl, clv)
- `srcip`: public source IP address from which the client connected
- `virtual_ip`: private pool IP address that was assigned to the user

Only `variant=clv`:

- `type: HTML5 type: VNCRDeskop | VNCWebapp | PTYSSH | PTYTelnet | VNC`

```
2009:09:15-10:57:58 utm pppd-pptp[5467]:  id="2202" severity="info"
sys="SecureNet" sub="vpn" event="Connection terminated"  username="bertram"
variant="pptp" srcip="192.168.2.96" virtual_ip="10.242.1.2"
```

Chapter 8

WiFi

Logfiles these events can be found in are:

- `wireless.log`

STA connected

Description: This log entry is being generated when a wireless STA establishes a connection with an AP (authentication, association, key negotiation and EAP are finished successfully)

ID: 4101

- `ssid`: The SSID the STA connected to
- `ssid_id`: The ID of the wireless network the STA connected to and the Band (0=2,4 GHz, 1=5 GHz) seperated with a dot
- `bssid`: MAC address of the virtual AP
- `sta`: MAC address of the connecting STA

Jan 12 14:32:21 OpenWrt user.notice awelogger[1385]: id="4101" severity="info" sys="System" sub="WiFi" name="STA connected" ssid="WPAEAPTest" ssid_id="WLAN1.0" bssid="00:11:22:33:44:56" sta="00:11:22:33:44:55"

STA disconnected

Description: A STA disconnected from an AP

ID: 4102

- `ssid`: The SSID the STA was connected to
- `ssid_id`: The ID of the wireless network the STA was connected to and the Band (0=2,4 GHz, 1=5 GHz) seperated with a dot
- `bssid`: MAC address of the virtual AP
- `sta`: MAC address of the disconnecting STA

Jan 12 14:32:16 OpenWrt user.notice awelogger[1385]: id="4102" severity="info" sys="System" sub="WiFi" name="STA disconnected" ssid="WPAEAPTest" ssid_id="WLAN1.0"

bssid="00:11:22:33:44:56" sta="00:11:22:33:44:55"

STA authenticating

Description: A STA authenticated (802.11 auth, not EAP) to the AP, the status_code (as defined in 802.11-2007, 7.3.1.9, page 93) indicates success (=0) or failure (!=0)

ID: 4103

- `ssid`: The SSID the STA was connected to
- `ssid_id`: The ID of the wireless network the STA was connected to and the Band (0=2,4 GHz, 1=5 GHz) seperated with a dot
- `bssid`: MAC address of the virtual AP
- `sta`: MAC address of the disconnecting STA
- `status_code`: 802.11 status code

2011:04:04-10:23:41 10.254.0.11 awelogger[1863]: id="4103" severity="info" sys="System" sub="WiFi" name="STA authentication" ssid="AstaroTestSep" ssid_id="WLAN4.0" bssid="00:02:6f:75:4e:e4" sta="00:24:d6:72:aa:46" status_code="0"

STA associating

Description: A STA associated to the AP, the status_code (as defined in 802.11-2007, 7.3.1.9, page 93) indicates success (0) or failure (!=0)

ID: 4104

- `ssid`: The SSID the STA was connected to
- `ssid_id`: The ID of the wireless network the STA was connected to and the Band (0=2,4 GHz, 1=5 GHz) seperated with a dot
- `bssid`: MAC address of the virtual AP
- `sta`: MAC address of the associating STA
- `status_code`: 802.11 status code

2011:04:04-10:23:41 10.254.0.11 awelogger[1863]: id="4104" severity="info" sys="System" sub="WiFi" name="STA association" ssid="AstaroTestSep" ssid_id="WLAN4.0" bssid="00:02:6f:75:4e:e4" sta="00:24:d6:72:aa:46" status_code="0"

STA WPA failure

Description: A STA wasn't able to establish a WPA connection. The STA is most likely not using the correct PSK. The reason code field is defined by 802.11-2007, 7.3.1.7, page 92.

ID: 4105

- `ssid`: The SSID the STA was connected to
- `ssid_id`: The ID of the wireless network the STA was connected to and the Band (0=2,4 GHz, 1=5 GHz) seperated with a dot
- `bssid`: MAC address of the virtual AP

- `sta`: MAC address of the causing STA
- `reason_code`: 802.11 reason code

2011:04:04-10:23:44 10.254.0.11 awelogger[1863]: id="4105" severity="info" sys="System" sub="WiFi" name="STA WPA failure" ssid="AstaroTestSep" ssid_id="WLAN4.0" bssid="00:02:6f:75:4e:e4" sta="00:24:d6:72:aa:46" reason_code="2"