- session_timeout	is the amount of time (in seconds) the customer is allowed on the network.
- idle_timeout	is the amount of time (in seconds) the customer can be idle before being logged out.
- interim_interval	is the accounting interim interval (in seconds) for this user.
- bandwidth_max_up	is the maximum upload bandwidth the user is allowed to consume on this Xnet Viper, specified in kilobits per second.
- bandwidth_max_down	is the maximum download bandwidth the user is allowed to consume on this Xnet Viper, specified in kilobits per second.
- max_sessions	is the maximum number of simultaneous TCP/IP sessions a user may use. Reasonable sane values are 100 to 250. Use a value of zero to disable session checking for a particular user.
- USER_TYPE	Is the type of user that is being logged in. This can be: **web**, normal user browsing the internet. **voip**, user with a VoIP device, such as WiFi telephone. **virus**, user that has been placed in a restricted area because of being infected by a virus. **other**, user definable queue.

If the optional parameters are not specified, the values define in the configuration files are used instead. In software version 3.0.x those parameters are set to:

- session_timeout	14400 (4 hours)
- idle_timeout	600 (10 minutes)
- interim_interval	300 (5 minutes)
- bandwidth_max_up	10240 (10 Megabits per second)
- bandwidth_max_down	10240 (10 Megabits per second)
- max_sessions	100
- USER_TYPE	web

If we continue with the example from message 1 then the authentication response from the back office would be:

```
http://10.16.0.15/authorizeUser.php?action=open&cust_IP=192.168.2.10&cust_MAC=00:02:
2D:42:4B:1C&session_timeout=3600&idle_timeout=600&bandwidth_max_up=128&bandwidth_max
_down=512
```

10.2.5 Xnet Viper XML responses (message 3)

When the back office calls the authorizeUser.php page on the Xnet Viper, the Xnet Viper will respond to the request with an answer in XML code. The response indicates whether or not the action was correctly performed. Below is on overview of the possible responses.

10.2.5.1 Response: OK, Successful user authorization

Below is an example of a response to a successful user authorization:

```
<?xml version="1.0" encoding="utf-8" ?>
<authorizeUser>
<action>open</action>
   <okay>
      <code>0</code>
      <message>Login successful</message>
   </okay>
   <cust_MAC>00:0E:35:91:27:E2</cust_MAC>
   <cust_IP>192.168.0.250</cust_IP>
   <public_ip></public_ip>
   <cust_VGW>0</cust_VGW>
   <cust_TYPE>web</cust_TYPE>
```

```
   <cust_name></cust_name>
   <cust_password> </cust_password>
   <session_timeout>14400</session_timeout>
   <session_start>1176892081</session_start>
   <session_end>0</session_end>
   <idle_timeout>600</idle_timeout>
   <interim_interval>300</interim_interval>
   <bytes_up>0</bytes_up>
   <bytes_down>0</bytes_down>
   <bandwidth_max_up>10240</bandwidth_max_up>
   <bandwidth_max_down>10240</bandwidth_max_down>
   <max_sessions>100</max_sessions>
   <cur_sessions>0</cur_sessions>
   <enable_accounting>0</enable_accounting>
   <status>OK</status>
</authorizeUser>
```

10.2.5.2 Response: Error code 2, User MAC address is not specified

When the back office does not specify the mandatory MAC address in the http GET operation to the Xnet Viper then the following error is generated:

```
<?xml version="1.0" encoding="utf-8" ?>
<authorizeUser>
   <error>
      <code> 2 </code>
      <message>User MAC address is not specified</message>
   </error>
</authorizeUser>
```

10.2.5.3 Response: Error code 3, User IP address is not specified

When the back office does not specify the mandatory IP address in the http GET operation to the Xnet Viper the following error is generated:

```
<?xml version="1.0" encoding="utf-8" ?>
<authorizeUser>
   <error>
      <code> 3 </code>
      <message>User IP address is not specified</message>
   </error>
</authorizeUser>
```

10.2.5.4 Response: Warning code 5, Virtual Gateway number is not specified. Assume 0

When the portal tries to log a user in it communicates with the Xspot through the VPN tunnel. With the introduction of the Virtual Gateway concept an Xspot can have 4 different Virtual Gateways and 4 different VPN tunnels. The gateways are numbered zero to three. Every Virtual Gateway has its own VPN tunnel associated with it. Authorize_user requests coming in through the VPN tunnel of a particular Virtual Gateway are only able to log users in and out for that Virtual Gateway. So it is impossible for one operator to log users in and out of another operators' Virtual Gateway. The Virtual Gateway number is calculated from the number of the VPN tunnel. In case an error occurs and the Virtual Gateway number is not correctly calculated a warning is generated and Virtual Gateway 0 is assumed.

Below is an example of the output when the Virtual Gateway is not specified.

```
<authorizeUser>
   <action>open</action>
   <cust_MAC>00:02:2d:42:4b:1c</cust_MAC>
   <cust_IP>192.168.0.11</cust_IP>
   <warning>
      <code> 5 </code>
      <message>Virtual Gateway number is not specified. Assume 0.</message>
   </warning>
   <cust_VGW>0</cust_VGW>
   <cust_TYPE>web</cust_TYPE>
   <cust_name></cust_name>
   <cust_password></cust_password>
```

```
   <session_timeout>14400</session_timeout>
   <session_start>1125773753</session_start>
   <idle_timeout>600</idle_timeout>
   <interim_interval>300</interim_interval>
   <bandwidth_max_up>10240</bandwidth_max_up>
   <bandwidth_max_down>10240</bandwidth_max_down>
   <max_sessions>100</max_sessions>
   <cur_sessions>0</cur_sessions>
   <enable_accounting>no</enable_accounting>
   <status>OK</status>
</authorizeUser>
```

During normal operation of the system this warning should never appear.

10.2.5.5 Response: Warning code 6, User type is not specified. Assume web

The Xspot gateway now has 4 queues that can be used to log users in and out. Normally users would be logged in the web queue. However, users can also be logged in the **voip**, **virus** and **other** queue. This queue has to be specified by the portal when it calls the authorize_user.php URL on the Xnet Viper by means of the USER_TYPE parameter. When the USER_TYPE parameter is not specified the Xnet Viper assumes it is USER_TYPE web and generates a warning.
Below is an example of the output when the user type is not specified:

```
<?xml version="1.0" encoding="utf-8" ?>
<authorizeUser>
   <action>open</action>
   <cust_MAC>00:02:2d:42:4b:1c</cust_MAC>
   <cust_IP>192.168.0.11</cust_IP>
   <cust_VGW>0</cust_VGW>
   <warning>
      <code> 6 </code>
      <message>User type is not specified. Assume web.</message>
   </warning>
   <cust_TYPE>web</cust_TYPE>
   <cust_name></cust_name>
   <cust_password></cust_password>
   <session_timeout>14400</session_timeout>
   <session_start>1125773753</session_start>
   <idle_timeout>600</idle_timeout>
   <interim_interval>300</interim_interval>
   <bandwidth_max_up>10240</bandwidth_max_up>
   <bandwidth_max_down>10240</bandwidth_max_down>
   <max_sessions>100</max_sessions>
   <cur_sessions>0</cur_sessions>
   <enable_accounting>no</enable_accounting>
   <status>OK</status>
</authorizeUser>
```

Currently, users should only be logged in into the web queue, as the other queues are reserved for future use.

10.2.6 Logging customers off

Customers can be logged off in three several ways.

1. They can be actively logged off by the back office by sending the "close" command to the Xnet Viper.
2. Customers have been idle for a longer period than the idle-timeout.
3. Customer's session time has expired.

10.2.7 Customer log off by the back office

In order to log the customer off the back office can execute an HTTP GET to the requesting Xnet Viper by calling the URL: http://tunnel_ip/authorizeUser.php through the VPN tunnel. This is an extension of message number 2.

Message 2 must contain the following mandatory information:

- tunnel_IP is the IP address of the VPN tunnel end point on the Hopling.
- action can be either "open" or "close". In this case close
- cust_MAC is the MAC address of the customer's device.
- cust_IP is the IP address of the customer's device.

When the back office calls the authorizeUser.php page on the Xnet Viper-I, the Xnet Viper will respond to the request with an answer in XML code. The response indicates whether or not the action was correctly performed. Below is on overview of the possible responses.

10.2.7.1 Response: OK, Successful user log off

Below is an example of a response to a successful user log off:

```
<?xml version="1.0" encoding="utf-8" ?>
<authorizeUser>
<action>close</action>
   <okay>
      <code>0</code>
      <message>Logout successful</message>
   </okay>
   <cust_MAC>00:0E:35:91:27:E2</cust_MAC>
   <cust_IP>192.168.0.250</cust_IP>
   <public_ip></public_ip>
   <cust_VGW>0</cust_VGW>
   <cust_TYPE>web</cust_TYPE>
   <cust_name></cust_name>
   <cust_password> </cust_password>
   <session_timeout>86400</session_timeout>
   <session_start>1176891879</session_start>
   <session_end>1176891999</session_end>
   <idle_timeout>600</idle_timeout>
   <interim_interval>300</interim_interval>
   <bytes_up>4341</bytes_up>
   <bytes_down>63312</bytes_down>
   <bandwidth_max_up>10240</bandwidth_max_up>
   <bandwidth_max_down>10240</bandwidth_max_down>
   <max_sessions>100</max_sessions>
   <cur_sessions>3</cur_sessions>
   <enable_accounting>0</enable_accounting>
   <status>OK</status>
</authorizeUser>
```

10.2.7.2 Response: Error code -1, Client not found

Below is an example of a response to an unsuccessful user log off. This situation can arise when the back office tries to log a user off that has already been logged off by either the idle timer or the session timer:

```
?xml version="1.0" encoding="utf-8" ?>
<authorizeUser>
<action>status</action>
   <error>
      <code>-1</code>
      <message>Client not found</message>
   </error>
</authorizeUser>
```

10.2.8 Customer log off due to idle time out

When the customer is idle for a longer period than the idle timer then the customer will automatically be logged off. The idle timeout is measured by looking at the amount of data sent through the wireless interface. Every time data is transferred between the customer's device and the Xnet Viper the idle timer will be reset. It can happen that customers have closed their browsers and other application but that the customer's device is still sending and receiving data. In that case the customer is not logged off.

The idle timeout can be specified on a per customer basis when authorizing users. If the optional **`idle_timeout`** parameter is not specified then the default value from the `/config/hopling/virtual_gw/virtual_gw_0/hotspot_mode/hotspot.conf` is used. See also §10.2.4 User Authentication (message 2).

10.2.9 Customer log off due to session time out

When the customer is active for a longer period than specified by the optional session timeout the customer will automatically be logged off. The session timeout can be specified on a per customer basis when authorizing users. If the optional **`session_timeout`** parameter is not specified then the default value from the `/config/hopling/virtual_gw/virtual_gw_0/hotspot_mode/hotspot.conf` is used. See also §10.2.4 User Authentication (message 2).

11 Xnet logging system events to a central web portal

The Xnet is capable of logging a number of system events to a central web server or portal. This can be useful for monitoring the Xnet nodes in the network or simply storing the events in a database. Each Virtual gateway is capable of sending a set of events to a separate back office.

Events are usually sent when the state of the system or subsystem changes. For example when the system is switched on it will generate a "boot" event to signal to the central portal that it has just become active. Other events can be triggered by connecting and disconnecting customers and by external systems such as access points.

Events are sent to the portal by means of an "http GET" operation. When an event is triggered the Xnet will contact the central portal web server on a predefined URL. This predefined portal server and logging URL are set in the Virtual Gateway configuration files of the Xnet. The file

/config/hopling/virtual_gw/virtual_gw_0/ virtual_gw.0.conf

contains the logging server and the logging URL. Because the events are fully configurable through the event configuration file it is important to understand the format of the event URL. The URL is composed of a number of items.

- LOG_SERVER (configurable)
- LOG_URL (configurable)
- Unix date and Hopling string (fixed)
- Event name (configurable)
- General Xnet Viper parameters (configurable)
- Specific parameters different for each event (configurable)

The format of the http GET operation is:

```
http://www.hopling.nl/download_config/log.php?event=eventname&date=
20050902145559&hopling= XnetMkII-c3a124&parameter1=parameter1&parameter2=parameter2
```

The parameters that are sent with each event are configurable through configuration files. Each event has a corresponding configuration file with the parameter names and the value that will be filled in for that parameter. The event files are found in each Virtual Gateway directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events
```

Most events, with the exception of the "reDirect", "idleTimeout", "sessionTimeout" and "virusDetect" events are system wide events. In other words they are sent to one portal web server regardless of the Virtual Gateways that are generating the events. Only the user specific "reDirect", "idleTimeout", "sessionTimeout" and "virusDetect events are sent to different portals connected to the different Virtual Gateways.

11.1 Setting up event logging to a central web portal.

Event monitoring begins with setting up the correct logging server and logging URL in the file: **/config/hopling/virtual_gw/virtual_gw_0/ virtual_gw.0.conf**. The parameter **VGW_0_LOG_SERVER** holds the IP address or hostname of the logging server and if prefixed with either http:// or https://. The VGW_0_LOG_URL is the location on the server that is called for each event.

For example the default values for the logging server and logging URL are set to:

```
#@$ <"Log parameters">
#@@ <VGW_0_LOG_SERVER>  <HOST_IP>   <1>              <NONE>   <RESERVED>  <"Server
to use for logging events">
```

```
#@@ <VGW_0_LOG_URL>     <URL>       <1>              <NONE>   <RESERVED>  <"URL to
call to log events through XML">

VGW_0_LOG_SERVER="http://hopman.hopling-services.net"
VGW_0_LOG_URL="hopman/log.php"
```

The parameters for each event are set in the event files. Every event has a corresponding parameter file. You can either change the parameters by logging in to the web interface or edit the files directly through the SSH shell and vi editor.

11.2 Event Overview

The following table gives an overview of each event that can be generated by the Xnet Xnet Viper and the default parameters that are passed to the portal for each event.

Event name	Purpose	Default parameters	Generated
11.2.2 apDetect	Report access points connected to the eth1 port of the Xnet Viper at regular intervals.	gateway_ID software_VER platform_VER platform_TYPE macEth0 ipEth0 dhcpClient requestId	When a new access point is connected to the Xnet Viper. Every 10 minutes if the access point stays connected and is operational.
11.2.3 gatewayDetect	Report the Xnet Viper being live at regular intervals	gateway_ID software_VER platform_VER platform_TYPE macEth0 ipEth0 macEth1 ipEth1 macWlan0 ipWlan0 wireless_NET vpn_server tunnel_ip	Every 10 minutes as long as the system is up.
11.2.4 getConfig	To start downloading the configuration files through the means of XML. See also chapter 12 Xnet call home functionality and XML interface	gateway_ID software_VER platform_VER macEth0 ipEth0 macEth1 ipEth1 macWlan0 ipWlan0	At system boot in case the CONFIG_METHOD="auto"
11.2.5 hoplingBoot	To signal to the portal that the Xnet system has just booted	gateway_ID software_VER platform_VER platform_TYPE macEth0 ipEth0 macEth1 ipEth1 macWlan0 ipWlan0 wireless_NET vpn_server tunnel_ip	Always sent at system boot
11.2.6 idleTimeout	To signal to the portal that a user has been logged out due to an idle timeout	gateway_ID software_VER platform_VER platform_TYPE macEth0 wireless_NET cust_MAC cust_IP	When a user is logged out due to idle timeout
11.2.7 pubIpChanged	To signal that the Xnet Viper received a new external IP address on its WAN port	gateway_ID software_VER platform_VER platform_TYPE macEth0 ipEth0 macEth1 ipEth1 macWlan0 ipWlan0 wireless_NET vpn_server tunnel_ip	When the WAN Ethernet port receives a new and different IP address from an external DHCP server
11.2.8	To signal to the portal that a	gateway_ID	When a user is logged out due

sessionTimeout	user has been logged out due to a session timeout	software_VER platform_VER platform_TYPE macEth0 wireless_NET cust_MAC cust_IP	to session timeout
11.2.9 reDirect	To redirect the user's browser to the login portal	gateway_ID software_VER platform_VER platform_TYPE cust_MAC cust_IP cust_URL cust_DHCP public_IP private_IP tunnel_IP	Every time a user opens his web browser and is not logged in on the Xnet Viper. When the user tries to access an arbitrary URL his browser will be redirected to the loin portal
11.2.10 uploadConfig	To upload the current configuration of the Xnet Viper to the configuration portal server	gateway_ID software_VER platform_VER macEth0 ipEth0 macEth1 ipEth1 macWlan0 ipWlan0	Every time the configuration server requests to upload the Xnet Viper's configuration. This can be after the "hoplingBoot" event or by calling the URL: uploadConfig.php through the VPN tunnel.
11.2.11 virusDetect	To signal to the portal that a user is generating an excessive amount of TCP/IP sessions	gateway_ID software_VER platform_VER platform_TYPE macEth0 wireless_NET cust_MAC cust_IP	When a user generates more than the predefined number of TCP/IP sessions per minute. The exact amount can be set during authentication of the user. When not set the default value from the configuration file is used. See paragraph 10.2.4

11.2.1 General Event parameters

Every event that is sent to the remote portal contains a number of general parameters and a number of parameters that are specific for that event. For example the apDetect event would contain the GATEWAY_ID to uniquely identity the Xnet Viper that is generating the event to the portal. It would also contain the MAC address of the access point. Whereas the GATEWAY_ID is present in every event, the access point MAC address is only present in the apDetect event.

Below is a list of general parameters that can be sent with every event:

General Parameters	Description
$HOST_NAME	The host name of the Xnet in case it is set in the configuration file. Otherwise this parameter is empty.
$LOCATION_ID	The location ID of the Xnet in case it is set in the configuration file. Otherwise this parameter is empty.
$GATEWAY_ID	The gateway ID of the Xnet in case it is set in the configuration file. Otherwise this parameter is empty.
$CLIENT_STRING	The ID of the Xnet. For example: XnetMkI-c20930
$HW_TYPE	The type radio hardware this software is running on. Possible values are: - **PRISMWWR**, 1 radio card: Prism World radio hardware. - **PRISM2**, 1 radio card: Prism2/2.5/3 radio hardware. - **ATH**, 1 radio card: Atheros radio hardware. - **PRISM2_PRISMWWR**, 2 radio cards: Prism 2 and Prism World Radio hardware. - **PRISM2_ATH**, 2 radio cards: Prism2 and Atheros hardware. - **AR-M9926_NO_WIFI**, Xfire platform and no Wifi radio hardware. - **NET4801_NO_WIFI**, Net4801 platform and no Wifi radio hardware. - **OPTION**, UMTS radio card type Option and no Wifi radio hardware. - **OPTION_PRISMWWR**, UMTS radio card type Option and Prism World radio hardware. - **OPTION_ATH**, UMTS radio card type Option and Atheros radio hardware. - **unknown**, if the software is running with an unknown (radio) hardware configuration.
$SW_VERSION	The software version that the Xnet Viper is running. For example: 3.0.1
$HW_PLATFORM	The hardware platform containing the radio cards and running the software. Possible values are: - **Net4511**, a 100MHz AMD Elan with two 100Mbps Ethernet ports, 1 mini-PCI slot and 1 PCMCIA slot. - **Net4521**, a 100MHz AMD Elan with two 100Mbps Ethernet ports, 1 mini-PCI slot and 2 PCMCIA slots. - **Net4801**, a 366MHz Geode processor with three 100Mbps Ethernet ports, 1 mini-PCI slot and 1 PCI slot - **MS2100**, Geode processor with one 100Mbps Ethernet port and 1 PCMCIA slot (obsolete). - **AR-M9926**, 1GHz Geode processor with four 100Mbps Ethernet ports and 1 PCI slot. - **unknown**, if the software is running on a platform that is not recognized by the software.
$BUILD_TAG	The build tag that is assigned to the software running on the Xnet. This is a qualification that is automatically generated for every software build that is released by Hopling Technologies. It should only contain the tag "release". For Hopling Technologies internal software development purposes it may contain "development".
$BUILD_NR	The build number that is assigned to the software running on the Xnet. This is a sequential number that is automatically generated for every software build that is released by Hopling Technologies.
$FLAVOUR	The FLAVOUR parameter is to indicate the customer for which this build was made. Customers can, upon request, get their own "flavour" of the Xnet software. Normally this parameter will be set to "HT1".
$FLAVOUR_TYPE	Xnet or Xspot

11.2.2 Event: apDetect

The apDetect event is generated every time the Xnet Viper detects that an access point is connected to the eth1 port. This feature is disabled by default. It can be enabled by setting parameter: `ENABLE_AP_MON="yes"` in the `/config/hopling/hopling.conf` configuration file.

When the apDetect feature is enabled new access points are automatically reported to the portal web server belonging to Virtual Gateway 0 (zero). The portal server can request to start monitoring the access points by sending back the MAC address of the access point device using XML code. If the Xnet access controller needs to start monitoring the device it will send the apDetect event every interval as specified by the parameter: `AP_DETECT="600"`. The default value for this parameter is set to 10 minutes (600 seconds).

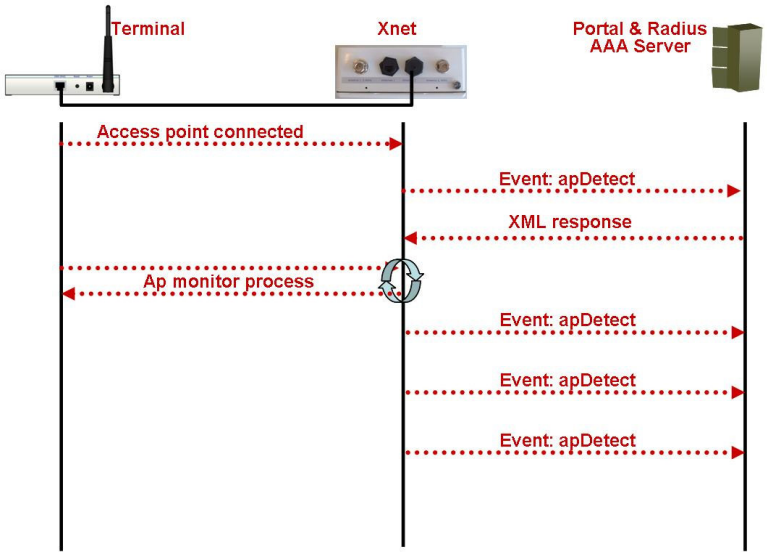

Figure 30 apDetect event

The following access points are automatically detected by the Xnet Viper:

- Hopling Xlite
- Cisco 1200
- Cisco 1100
- Colubris CN320
- Colubris CN200

The parameters sent in the apDetect event can be configured in the corresponding apDetect event file. The event file is found in the directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events.
```

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the apDetect event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/apDetect
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the "apDetect" event.
#
# Event name
#TITLE="configuration parameters for the apDetect event."
#START
event apDetect

# Additional parameters
gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
ipEth0 $BR_WAN0_IP
dhcpClient $AP_DEVICE_ID
requestId 0
```

When an apDetect is generated with the above configuration for the apDetect event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/log.php?date=20050903222701&hopling=XnetMkI-
c20930&event=apDetect&gateway_ID=&software_VER=3.01&platform_VER=Net4511&platform_TYPE=PRISMWWR&
software=3.01&platform=Net4511&macEth0=00:12:80:0d:43:bc&ipEth0=192.168.10.20&dhcpClient=SIP0012
800D43BC&requestId=0
```

The apDetect event can contain a large number of additional parameters. Below is a list of parameters that are specific for the apDetect event that can be used to customize the apDetect event:

AP specific Parameters	Description
$AP_DEVICE_ID	The ID of the connected access point, for example: Hopling_Cisco_1200
$AP_MAC_ADDRESS	The MAC address of the connected access point, for example: 00:12:80:0d:45:6c
$AP_IP_ADDRESS	The IP address of the connected access point, for example: 192.168.10.22
$AP_PORT_ID	The port ID of the Ethernet port of the connected access point, for example: FastEthernet0
$AP_SW_VERSION	The software version of the connected access point. For example: Cisco Internetwork Operating System Software IOS (tm) C1200 Software (C1200-K9W7-M)
$AP_HW_PLATFORM	The hardware platform of the connected access point. For example: Cisco AIR-AP1231G-E-K9
$AP_DETECT_MODE	The way the access point was detected by the Xnet Viper. This can either be **cdp** or **dhcp**
$AP_MONITORING	Whether or not this access point is being monitored by the Xnet Viper. Values can be: **yes** or **no**. During the discovery stage of the access point this value will be empty.
$AP_FIRST_SEEN	The Unix system time when the access point was first seen. This is generally the time that the access point was first booted and/or connected. For example: 1125675708
$AP_LAST_SEEN	The Unix system time when the access point was last seen. This is the time that the access point last reported to the Xnet Viper. For example: 1125675888
$AP_LAST_REPORTED	The UNIX system time when the access point was last reported to the central web portal. For example: 1125675940

11.2.2.1 Signaling to start monitoring an Access Point

The portal server can signal to the Xnet Viper to start monitoring the access point after it receives an apDetect event. This is done by sending back the access point MAC address in the response to the http call. The format of the response is in XML.

```
<?xml version="1.0" encoding="utf-8" ?>
<hopling>
   <ap>
      <mac>00:12:80:0d:43:bc</mac>
   </ap>
</hopling>
```

When the Xnet Viper receives the XML answer with the MAC address of the access point after it has sent the apDetect message to the portal it will start monitoring the access point.

11.2.3 Event: gatewayDetect

The gatewayDetect event is meant to signal to the remote portal website that the Xnet Viper is still alive. This feature is disabled by default. It can be enabled by setting parameter: `ENABLE_GW_MON="yes"` in the `/config/hopling/hopling.conf` configuration file.

When the gatewayDetect feature is enabled the Xnet Viper will automatically generate the event every 10 minutes (600 seconds). The event interval can be changed by setting the value of the parameter: `GW_DETECT="600"`. The default value of this parameter is set to 10 minutes (600 seconds).

The parameters sent in the gatewayDetect event can be configured in the corresponding gatewayDetect event file. The event file is found in the directory:

`/config/hopling/virtual_gw/virtual_gw_0/events.`

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the gatewayDetect event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/gatewayDetect
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the "gatewayDetect" event.
#
# Event name
#TITLE="configuration parameters for the gatewayDetect event."
#START
event gatewayDetect

# Additional parameters
gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
ipEth0 $BR_WAN0_IP
macEth1 $MAC_ETH1
ipEth1 $BR_VGW0_IP
macWlan0 $WIFI_0_0_MAC
ipWlan0 $WIFI_0_0_IP
wireless_NET $WIFI_0_0_SSID
vpn_server $VGW_0_VPN_SERVER
tunnel_ip $VGW_0_VPN_LOC_IP
```

When a gatewayDetect is generated with the above configuration for the gatewayDetect event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/log.php?date=20050904134700&hopling=XnetMkI-
c20930&event=gatewayDetect&gateway_ID=&software_VER=3.0.1&platform_VER=Net4511&macEth0=00:00:24:
c2:09:30&ipEth0=192.168.10.202&macEth1=00:00:24:c2:09:31&ipEth1=192.168.0.1&macWlan0=00:0c:84:01
:31:8c&ipWlan0=&wireless_NET=Hopling Technologies 0
```

This event can only use the general event parameters. No additional parameters are available for this event.

11.2.4 Event: getConfig

The getConfig event is meant to signal to the remote portal website that the Xnet Viper is requesting that its configuration parameters are to be sent by means of XML. The event is generated once, just after the Xnet Viper has booted. How to generate the correct XML response syntax for this event is further described in chapter: 12.

This feature is enabled by default. It can be disabled by setting parameter: `CONFIG_METHOD="manual"` in the `/config/hopling/hopling.conf` configuration file.

The server and URL that are accessed during the getConfig event differ from the logging server and logging URL. This is done because the network operator might be running the logging functions and configuration functions on separate servers. The configuration server and configuration URL are defined in the configuration file: `/config/hopling/hopling.conf`. The following are the factory default values that are set:

```
CONFIG_SERVER="http://www.hopling.nl"
CONFIG_URL="download_config/factory_defaults.php"
```

The parameters sent in the getConfig event can be configured in the corresponding getConfig event file. The event file is found in the directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events
```

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the getConfig event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/getConfig
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration paramaters for the "getConfig" event.
#
# These are the Hopling Technologies parameters
# Event name
#TITLE="configuration parameters for the getConfig event."
#START
event getConfig

# Additional Hopling parameters
gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
ipEth0 $BR_WAN0_IP
macEth1 $MAC_ETH1
ipEth1 $BR_VGW0_IP
macWlan0 $WIFI_0_0_MAC
ipWlan0 $WIFI_0_0_IP
```

When a getConfig event is generated with the above configuration for the getConfig event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/factory_defaults.php?date=20050904141020&hopling=XnetMkI-
c20930&event=getConfig&gateway_ID=&software_VER=3.0.1&platform_VER=Net4511&macEth0=00:00:24:c2:0
9:30&ipEth0=192.168.10.202&macEth1=00:00:24:c2:09:31&ipEth1=192.168.0.1&macWlan0=00:0c:84:01:31:
8c&ipWlan0=&wireless_NET=Hopling Technologies 0
```

This event can only use the general event parameters. No additional parameters are available for this event.

11.2.5 Event: hoplingBoot

The hoplingBoot event is meant to signal to the remote portal website that the Xnet Viper has just booted. The event is generated once, just after time the Xnet Viper has booted. The remote portal server can request to upload the Xnet Viper's configuration in response to the hoplingBoot event.

The hoplingBoot event feature is enabled by default and it can not be disabled.

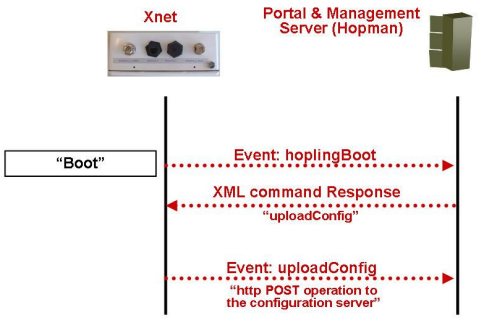

Figure 31, hoplingBoot event and subsequent uploadConfig event

The parameters sent in the hoplingBoot event can be configured in the corresponding hoplingBoot event file. The event file is found in the directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events
```

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the hoplingBoot event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/hoplingBoot
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006, 2007
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the "hoplingBoot" event.
#
# Event name
#TITLE="configuration parameters for the hoplingBoot event."
#START
event hoplingBoot

# Additional parameters
gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
ipEth0 $BR_WAN0_IP
macEth1 $MAC_ETH1
ipEth1 $BR_VGW0_IP
macWlan0 $WIFI_0_0_MAC
ipWlan0 $WIFI_0_0_IP
wireless_NET $WIFI_0_0_SSID
vpn_server $VGW_0_VPN_SERVER
```

```
tunnel_ip $VGW_0_VPN_LOC_IP

wifi_0_0_mac $WIFI_0_0_MAC
wifi_1_0_mac $WIFI_1_0_MAC
wifi_2_0_mac $WIFI_2_0_MAC
wifi_3_0_mac $WIFI_3_0_MAC
wifi_0_0_ssid $WIFI_0_0_SSID
wifi_1_0_ssid $WIFI_1_0_SSID
wifi_2_0_ssid $WIFI_2_0_SSID
wifi_3_0_ssid $WIFI_3_0_SSID

pcibus_0_systemid $PCIBUS_0_SYSTEMID
pcibus_0_subsystemid $PCIBUS_0_SUBSYSTEMID
pcibus_1_systemid $PCIBUS_1_SYSTEMID
pcibus_1_subsystemid $PCIBUS_1_SUBSYSTEMID
pcibus_2_systemid $PCIBUS_2_SYSTEMID
pcibus_2_subsystemid $PCIBUS_2_SUBSYSTEMID
pcibus_3_systemid $PCIBUS_3_SYSTEMID
pcibus_3_subsystemid $PCIBUS_3_SUBSYSTEMID
```

When a hoplingBoot event is generated with the above configuration for the hoplingBoot event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/log.php?date=20050904144718&hopling=XnetMkI-
c20930&event=hoplingBoot&gateway_ID=&software_VER=3.0.1&platform_VER=Net4511&platform_TYPE=PRISM
WWR&macEth0=00:00:24:c2:09:30&ipEth0=192.168.10.202&macEth1=00:00:24:c2:09:31&ipEth1=192.168.0.1
&macWlan0=00:0c:84:01:31:8c&ipWlan0=&wireless_NET=Hopling Technologies
0&vpn_server=82.148.221.165&tunnel_ip=169.254.255.10
```

This event can only use the general event parameters. No additional parameters are available for this event.

11.2.5.1 Calling of the uploadConfig event

The portal server can signal to the Xnet Viper to send the current configuration to the CONFIG server after it receives a hoplingBoot event. This is done by sending back an uploadConfig command in the response to the http GET call. The format of the response is in XML.

```
<?xml version="1.0" encoding="utf-8" ?>
<hopling>
   <command>uploadConfig</command>
</hopling>
```

When the Xnet Viper receives the XML answer with the command `uploadConfig` after it has sent the hoplingBoot message to the portal it will start uploading the Xnet Viper's configuration by means of an http POST command to the configuration server. The configuration server and configuration URL are defined in the configuration file:

```
/config/hopling/hopling.conf
```

The following are the factory default values that are set:

```
CONFIG_SERVER="http://www.hopling.nl"
CONFIG_URL="download_config/factory_defaults.php"
```

For a further description of the "uploadConfig" event refer to paragraph: 11.2.10.

11.2.6 Event: idleTimeout

The idleTimeout event is meant to signal to the remote portal website that a particular user has been logged out due to idle timeout. The event is sent to the portal that is associated with a particular virtual gateway. So if a user in Virtual Gateway 0 is idle, the idleTimeout event will be sent to the portal of operator 0.

The parameters sent in the idleTimeout event can be configured in the corresponding idleTimeout event file. The event file is found in the directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events.
```

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the idleTimeout event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/idleTimeout
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the "idleTimeout" event.
#
# Event name for Hopling
#TITLE="configuration parameters for the idleTimeout event."
#START
event idleTimeout

gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
wireless_NET $WIFI_0_0_SSID
cust_MAC $CUST_MAC
cust_IP $CUST_IP

bytes_up $BYTES_UP
bytes_down $BYTES_DOWN
packets_up $PACKETS_UP
packets_down $PACKETS_DOWN
cur_sessions $CUR_SESSIONS
```

When an idleTimeout is generated with the above configuration for the idleTimeout event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/log.php?date=20050904165121&hopling=XnetMkI-
c20930&event=idleTimeout&gateway_ID=&software_VER=3.0.1&platform_VER=Net4511&platform_TYPE=PRISM
WWR&macEth0=00:00:24:c2:09:30&wireless_NET=Hopling Technologies
0&cust_MAC=00:12:f0:a1:ad:a2&cust_IP=192.168.0.11
```

The idleTimeout event can contain a large number of additional parameters. Below is a list of parameters that are specific for the idleTimeout event that can be used to customize the idleTimeout event:

idleTimeout specific Parameters	Description
$CUST_MAC	The MAC address of the user's Wifi device, for example: 00:12:80:0d:45:6c
$CUST_IP	The IP address of the user's Wifi device, for example 192.168.0.11
$CUST_DHCP	The DHCP client name of the user's Wifi device if set by the operating system. For example: ivo's laptop.
$CUST_VGW	The Virtual Gateway the user is logged in to. Possible values 0 to 3 (zero to three).
$CUST_TYPE	The type of customer that is generating the idleTimeout event. Possible values are: - **web**, normal user browsing the internet. - **voip**, user with a VoIP device, such as WiFi telephone. - **virus**, user that has been placed in a restricted area because of being infected by a virus. - **other**, user definable queue.
$IDLE_TIMEOUT	The setting of the idle timeout value. Default value is set to 10 minutes (600 seconds).
$IDLE_TIME	The value of the idle timer. When the idle timeout occurs this timer will be set to the same value as the $IDLE_TIMEOUT
$SESSION_TIMEOUT	The setting of the session timeout value. Default value is set to 14400 seconds (4 hours).
$SESSION_TIME	The value of the session timer when the idle timeout occurs.
$INTERIM_INTERVAL	
$SESSION_START	The UNIX system time when the user's session started. For example: 1125675940
$BYTES_UP	The number of bytes the user has sent upstream since the beginning of the session.
$BYTES_DOWN	The number of bytes the user has received since the beginning of the session
$BANDWIDTH_MAX_UP	The maximum upload speed the user is allowed to use during this session. Specified in kilobits per second.
$BANDWIDTH_MAX_DOWN	The maximum download speed the user is allowed to use during this session. Specified in kilobits per second.
$MAX_SESSIONS	Number of concurrent TCP/IP sessions per minute the user is allowed to consume
$CUR_SESSIONS	Number of concurrent TCP/IP sessions per minute the used has at the time of the idle timeout event

11.2.7 Event: pubIpChanged

The pubIpChanged event is meant to signal to the remote portal website that the IP address of the upload or WAN port of the Xnet Viper has changed. An IP change can occur when the DHCP server giving out IP addresses for the WAN interface of the Xnet Viper decides to give a different IP address than the previous one. When an IP happens, the Xnet Viper has to restart its firewall and all user sessions will be logged out.

The parameters sent in the pubIpChanged event can be configured in the corresponding pubIpChanged event file. The event file is found in the directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events.
```

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the pubIpChanged event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/pubIpChanged
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the
# "pubIpChanged" event. This event occurs when the Xspot receives a
# new IP address from the DHCP server connected to the public Ethernet interface.
#
# Event name
#TITLE="configuration parameters for the pubIpChanged event."
#START
event pubIpChanged

# Additional parameters
gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
ipEth0 $BR_WAN0_IP
macEth1 $MAC_ETH1
ipEth1 $BR_VGW0_IP
macWlan0 $WIFI_0_0_MAC
ipWlan0 $WIFI_0_0_IP
wireless_NET $WIFI_0_0_SSID
vpn_server $VGW_0_VPN_SERVER
tunnel_ip $VGW_0_VPN_LOC_IP
```

When a pubIpchanged event is generated with the above configuration for the pubIpChanged event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/log.php?date=20050904185121&hopling=XnetMkI-
c20930&event=pubIpChanged&gateway_ID=&software_VER=3.0.1&platform_VER=Net4511&macEth0=00:00:24:c
2:09:30&ipEth0=192.168.10.202&macEth1=00:00:24:c2:09:31&ipEth1=192.168.0.1&macWlan0=00:0c:84:01:
31:8c&ipWlan0=&wireless_NET=Hopling Technologies
0&vpn_server=82.148.221.165&tunnel_ip=169.254.255.10
```

This event can only use the general event parameters. No additional parameters are available for this event.

11.2.8 Event: sessionTimeout

The sessionTimeout event is meant to signal to the remote portal website that a particular user has been logged out due to a session timeout. The event is sent to the portal that is associated with a particular virtual gateway. So if a user in Virtual Gateway 0 is idle, the sessionTimeout event will be sent to the portal of operator 0.

The parameters sent in the sessionTimeout event can be configured in the corresponding sessionTimeout event file. The event file is found in the directory:

```
/config/hopling/virtual_gw/virtual_gw_0/events.
```

The following are the factory default values that are set.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the sessionTimeout event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/sessionTimeout
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the "sessionTimeout" event.
#
# Event name for Hopling
#TITLE="configuration parameters for the sessionTimeout event."
#START
event sessionTimeout

gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
macEth0 $MAC_ETH0
wireless_NET $WIFI_0_0_SSID
cust_MAC $CUST_MAC
cust_IP $CUST_IP

bytes_up $BYTES_UP
bytes_down $BYTES_DOWN
packets_up $PACKETS_UP
packets_down $PACKETS_DOWN
cur_sessions $CUR_SESSIONS
```

When a sessionTimeout is generated with the above configuration for the sessionTimeout event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/log.php?date=20050904165121&hopling=XnetMkI-
c20930&event=sessionTimeout&gateway_ID=&software_VER=3.0.1&platform_VER=Net4511&platform_TYPE=PR
ISMWWR&macEth0=00:00:24:c2:09:30&wireless_NET=Hopling Technologies
0&cust_MAC=00:12:f0:a1:ad:a2&cust_IP=192.168.0.11
```

The sessionTimeout event can contain a large number of additional parameters. Below is a list of parameters that are specific for the sessionTimeout event that can be used to customize the sessionTimeout event:

sessionTimeout specific Parameters	Description
$CUST_MAC	The MAC address of the user's Wifi device, for example: 00:12:80:0d:45:6c
$CUST_IP	The IP address of the user's Wifi device, for example 192.168.0.11
$CUST_DHCP	The DHCP client name of the user's Wifi device if set by the operating system. For example: ivo's laptop.
$CUST_VGW	The Virtual Gateway the user is logged in to. Possible values 0 to 3 (zero to three).
$CUST_TYPE	The type of customer that is generating the sessionTimeout event. Possible values are: - **web**, normal user browsing the internet. - **voip**, user with a VoIP device, such as WiFi telephone. - **virus**, user that has been placed in a restricted area because of being infected by a virus. - **other**, user definable queue.
$IDLE_TIMEOUT	The setting of the idle timeout value. Default value is set to 10 minutes (600 seconds).
$IDLE_TIME	The value of the idle timer. When the idle timeout occurs this timer will be set to the same value as the $IDLE_TIMEOUT
$SESSION_TIMEOUT	The setting of the session timeout value. Default value is set to 14400 seconds (4 hours).
$SESSION_TIME	The value of the session timer when the session timeout occurs.
$INTERIM_INTERVAL	
$SESSION_START	The UNIX system time when the user's session started. For example: 1125675940
$BYTES_UP	The number of bytes the user has sent upstream since the beginning of the session.
$BYTES_DOWN	The number of bytes the user has received since the beginning of the session
$BANDWIDTH_MAX_UP	The maximum upload speed the user is allowed to use during this session. Specified in kilobits per second.
$BANDWIDTH_MAX_DOWN	The maximum download speed the user is allowed to use during this session. Specified in kilobits per second.
$MAX_SESSIONS	Number of concurrent TCP/IP sessions per minute the user is allowed to consume
$CUR_SESSIONS	Number of concurrent TCP/IP sessions per minute the used has at the time of the session timeout event

11.2.9 Event: reDirect

When a customer opens the browser and attempts to surf to any http or https website, the Xnet Viper redirects the browser to a specific portal page: the so-called portal push page. The parameters that are sent to the portal server are dependent on the redirect event file.

Every Virtual Gateway has its own reDirect event file. The redirect event files are found in: `/config/hopling/events/virtual_gw_x` (where x is 0 to 3). It is therefore possible to tailor each event to the specific operator needs.

The server and URL that are accessed during the reDirect event differ from the logging server and logging URL. This is done because the network operator might be running the redirect and login functions on separate servers. The redirect server and redirect URL are defined per Virtual Gateway in the configuration file(s): `/config/hopling/virtual_gw/virtual_gw_x/hotspot_mode/hotspot.conf` (where x indicates the virtual gateway number, 0 to 3)

The following are the factory default values that are set for the redirect server and redirect string:

```
REDIRECT_SERVER="http://www.hopling.nl"
REDIRECT_URL="download_config/default.php"
```

The following are the factory default values that are set for Virtual Gateway 0.

```
#@! <upload> <event> <reserved2> <reserved3> <reserved4>
#@$ <"Virtual Gateway 0: Event file for the reDirect event">
#
# file:/config/hopling/virtual_gw/virtual_gw_0/events/reDirect
#
# Configuration file for the Hopling Xspot
# (c) Hopling Technologies 2004, 2005, 2006
# Ivo van Ling (support@hopling.com)
#
# This file contains the configuration parameters for the "reDirect" event.
#
# The following parameters are being sent with a
# redirect to the remote web portal.
# These are the default Hopling Technologies
# parameters:
#TITLE="configuration parameters for the reDirect event."
#START
event reDirect

gateway_ID $CLIENT_STRING
software_VER $SW_VERSION
flavour $FLAVOUR
flavour_type $FLAVOUR_TYPE
build_nr $BUILD_NR
build_tag $BUILD_TAG
platform_VER $HW_PLATFORM
platform_TYPE $HW_TYPE
cust_MAC $CUST_MAC
cust_IP $CUST_IP
cust_URL $CUST_URL
public_IP $BR_WAN0_IP
private_IP $BR_VGW0_IP
tunnel_IP $VGW_0_VPN_LOC_IP
tunnel_ip $VGW_0_VPN_LOC_IP
```

When a reDirect is generated with the above configuration for the reDirect event the following URL is called from the Xnet Viper:

```
http://www.hopling.nl/download_config/default.php?gateway_ID=&software_VER=3.0.1&platform_VER=Ne
t4511&platform_TYPE=PRISMWWR&public_IP=192.168.1.170&private_IP=192.168.0.1&tunnel_IP=0.0.0.0&cu
```

```
st_MAC=00:12:F0:A1:AD:A2&cust_IP=192.168.0.10&cust_URL=http%3A%2F%2Fwww.nu.nl%2F&cust_DHCP=D4KF4
Q1J
```

The reDirect event can contain a number of additional parameters. Below is a list of parameters that are specific for the reDirect event that can be used to customize the reDirect event:

reDirect specific Parameters	Description
$CUST_MAC	The MAC address of the user's Wifi device, for example: 00:12:80:0d:45:6c
$CUST_IP	The IP address of the user's Wifi device, for example 192.168.0.11
$CUST_DHCP	The DHCP client name of the user's Wifi device if set by the operating system. For example: ivo's laptop.
$CUST_URL	The original URL the user indented to visit.