UPPSALA
UNIVERSITET

v1.2, May 26, 2016
Author: Jerker Nyberg von Below

SOP – Arptrack and other tools for computer network management at BMC

1 What is arptrack?

Arptrack is a collection of tools for collecting computer network data from several sources and presenting them to a system administrator via a webpage.

2 Purpose

The purpose of these tools was originally to fulfill requirements of being able to track down activity on the network. The responsible persons for a subnet has to know who is doing what. When unwanted activity occur, an IP-address and a point in time are usually the clues we have to finding out what happened.

3 Discussion

These are scripts that evolved over several years administrating the BMC computer network. They collect data needed to understand what is happening at the network, even where we have no direct control of the equipment involved.

At first there was only *arpwatch* which sent mail when it detected new computers or when computers changed their IP-address. Over time other interesting data were collected. Originally it was possible to interactively search for an IP- or MAC-address through the hierarchy of the network router and switches. This search was triggered when new computers were detected via *arpwatch*. This system was replaced by the current *porterwatch* which is instead traversing all switches and ports, collecting data of what equipment is connected where, making the current views possible.

At some point we thought that there could be some existing network monitoring software that could replace this. We ran *Observium* in parallel with *arptrack*, but could not provide the data we wanted in a simple way. Since then the university has introduced a new IPAM solution called *BlueCat* that may be set up to replace *porterwatch* but not the presentation of the physical location of the network socket.

4 Data collection

Data is stored in text files containing associative array with key-value pairs.

4.1 Data types and representation

- *IP* is an IP-address
- *Mac* is a MAC-address
- *TimeStamp* is a time stamp in POSIX time
- *Switch* is the name of a switch
- *SwitchPort* is a physical switch interface/port name
- *()* is a list, may be tuple, triple or more
- *ComputerName* is the name of a computer

- *Vlan* is the number of a VLAN
- *CrossConnect* is the cross connect cabinet ID
- *NetworkSocket* is the network socket ID
- → The arrow symbolizes a lookup in the data
- > The greater-than sign symbolize a key-value hash

4.2 **Arpwatch data** Collect data from the router ARP-table via SNMP

- *IP* > *Mac*
- *Mac* > *IP*
- *IP* > *TimeStamp*
- *Mac* > *TimeStamp*

4.3 **Switchport socket data** Collect from manual Excel docs the cross connect cabinet patch cable connections

- *(Switch, port)* > *(CrossConnect, NetworkSocket, Vlan)*

4.4 **Switchalias data** Manual list of alternative switch names

- *Switch* > *(Switch, Switch, …)*

4.5 **Computername data** Check the DHCP-server log for the computer names

- *Mac* > *ComputerName*

4.6 **Porterwatch data** Iterate all switches and collect data over SNMP on which ports are active and what MAC-address is using them.

- *(Switch, Port)* > *Vlan*
- *(Switch, Port)* > *TimeStamp*
- *Mac* > *Vlan*
- *Mac* > *Switch*
- *Mac* > *SwitchPort*
- *Mac* > *(Switch, SwitchPort)*
- *Mac* > *TimeStamp*

5 Presentation

5.1 http://net.bmc.uu.se/net/arptrack.cgi is a Python CGI-script that presents two different views. It may only be accessed inside UpUnet by the persons responsible at campus and departments active at BMC and some at IT-division and IRT.

5.2 **Last seen IP**

This view is listing all IPs in a range and the current data found for them, looking up last seen time stamps, VLAN numbers, switch ports all the way to the network sockets. The data is selected in steps starting from the IP, pickup the current Mac-address, finding what Switch and Switch.

IP → *TimeStamp, Mac* → *ComputerName, (Switch, SwitchPort)* → *TimeStamp, Vlan, CrossConnect, NetworkSocket*

- The view may be used to find unused IPs, or IPs than can be used again, by sorting the table columns on the date where the IP was last seen.

- The view may be used to find physically in what network socket an IP is located

237	130.238.39.236	5c:26:a:67:c6:81	2015-11-25 15:50:20	C7:2-1	Gi1/0/52	2015-11-16 15:42:21	680	FBV-59MS1Q1	-	-
238	130.238.39.237	8:0:27:f5:b9:85	2015-11-24 13:34:18	C2960S-C5-3_3	Gi1/0/28	2015-03-05 12:53:36	680	FBV-VBOX-W10-TH	C5-D301-36-07	B5.414:4
239	130.238.39.238	8:0:27:f5:b9:85	2015-11-23 19:07:31	C2960S-C5-3_3	Gi1/0/28	2015-03-05 12:53:36	680	FBV-VBOX-W10-TH	C5-D301-36-07	B5.414:4
240	130.238.39.239	40:6c:8f:59:ea:a5	2015-11-24 20:25:06	C6:3	Fa0/41	2015-11-24 14:45:48	679	BMC-C02G9173DJYC	-	-

5.3 Last seen Vlan/Switch/SwitchPort/CrossConnect/Socket

This view lookup the above to show for each Switch and SwitchPort what Vlan, CrossConnect and Socket it is connected to. It also shows when it was last active and displays a list of Mac, IP and ComputerName that last were active there.
The *Switchport socket data* is added to the *Porterwatch data* meaning that network connections that have no detected activity are also shown.
Vlan → (Switch, SwitchPort) → Vlan, TimeStamp, ((Mac, IP, ComputerName), …), CrossConnect, NetworkSocket

- The view may be used to find all MAC-addresses on a Vlan, even when they currently have no usable IP-address, and find in what network socket they were last seen.
- The view may be used to see how long time ago the different switch ports on a switch last was in use. This is useful when planning the network infrastructure.
- The view may be used to network sockets with switch ports that never have been in use. Maybe these switch ports may be used for something else?
- This view may be used to find all network sockets that are connected to a certain Vlan.
- This view may show all connected network sockets in a corridor.
- This view may show all cross connect patch cables in a cabinet.

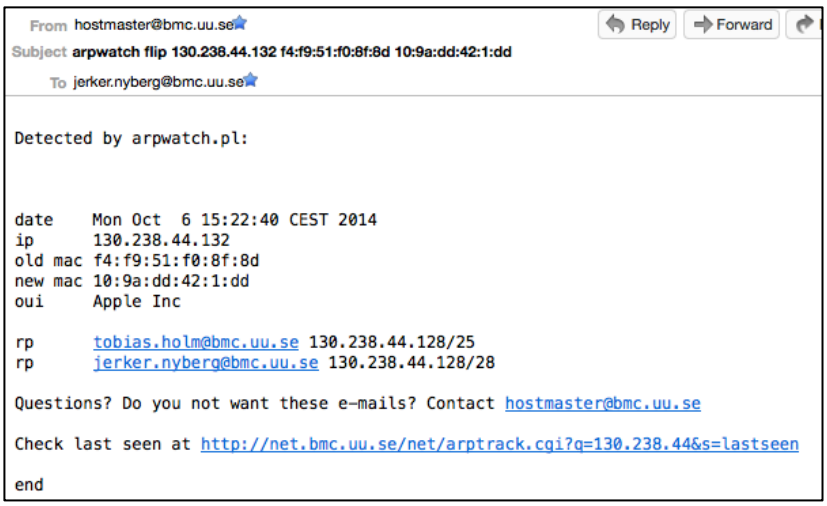
	Switch	Interface	Vlan	Interface last active	MAC IP Name	Cross connect ID	Socket ID
1	C2980:C5-3-2	3/10	680	2015-02-23 06:59:46		-	-
2	C7:3-5	Gi1/0/31	680	2015-01-27 08:30:53		C7-D301-21-20	C8.302b
3	C6:3	Fa0/28	680	2015-11-25 14:45:19	1 0:21:70:fa:cc:7d 130.238.39.214 BMC-4VGY94J	-	-
4	C2960S-C6:319c	Gi1/0/19	680	2015-11-25 15:33:59	1 0:1f:f3:46:18:50 130.238.39.248 -	-	-
5	C6:3	Fa0/11	680	2015-11-25 14:45:17	1 0:1a:a0:4d:54:15 130.238.39.245 BMC-99D403JW10	-	-
6	C2960S-C5-3_3	Gi1/0/28	680	2015-03-05 12:53:36		C5-D301-36-07	B5.414:4
7	C2960S-C6:319c	Gi1/0/25	680	2015-11-25 15:34:44	1 40:61:86:7d:88:95 130.238.39.207 bmc-oc1	-	-
8	C2960S-C6:319c	Gi1/0/24	680	2015-11-25 15:34:33	1 10:60:4b:92:a1:74 130.238.39.203 -	-	-
9	C2960S-C6:319c	Gi1/0/23	680	2015-11-25 15:34:29	1 8:0:27:4b:c9:1c 130.238.39.240 - 2 8:0:27:3a:80:46 130.238.39.235 - 3 8:0:27:22:de:c6 130.238.39.220 BMC-JERKERTW10 4 8:0:27:27:6:ad 130.238.39.229 BMC-JERKERTEST1 5 8:0:27:f8:b7:de 130.238.39.217 BMC-JERKERTEST2 6 a8:20:66:19:5b:b8 130.238.39.228 BMC-C07JD0NADJD3	-	-

6 Arpwatch

The first purpose of the arpwatch program was to send the responsible person for a subnet mail whenever an IP-address was changing MAC-address. The first time a certain IP- or MAC-address showed up was noted as new activity. The responsible person was

stored in RP-records in DNS. Currently, since the RP-records are gone, no mail is being sent.

- Useful for discovering possible undesirable network usage.
- Useful for discovering IP-address conflicts.



From hostmaster@bmc.uu.se
Reply Forward
Subject **arpwatch flip 130.238.44.132 f4:f9:51:f0:8f:8d 10:9a:dd:42:1:dd**
To jerker.nyberg@bmc.uu.se

```
Detected by arpwatch.pl:



date    Mon Oct  6 15:22:40 CEST 2014
ip      130.238.44.132
old mac f4:f9:51:f0:8f:8d
new mac 10:9a:dd:42:1:dd
oui     Apple Inc

rp      tobias.holm@bmc.uu.se 130.238.44.128/25
rp      jerker.nyberg@bmc.uu.se 130.238.44.128/28

Questions? Do you not want these e-mails? Contact hostmaster@bmc.uu.se

Check last seen at http://net.bmc.uu.se/net/arptrack.cgi?q=130.238.44&s=lastseen

end
```

UPPSALA
UNIVERSITET

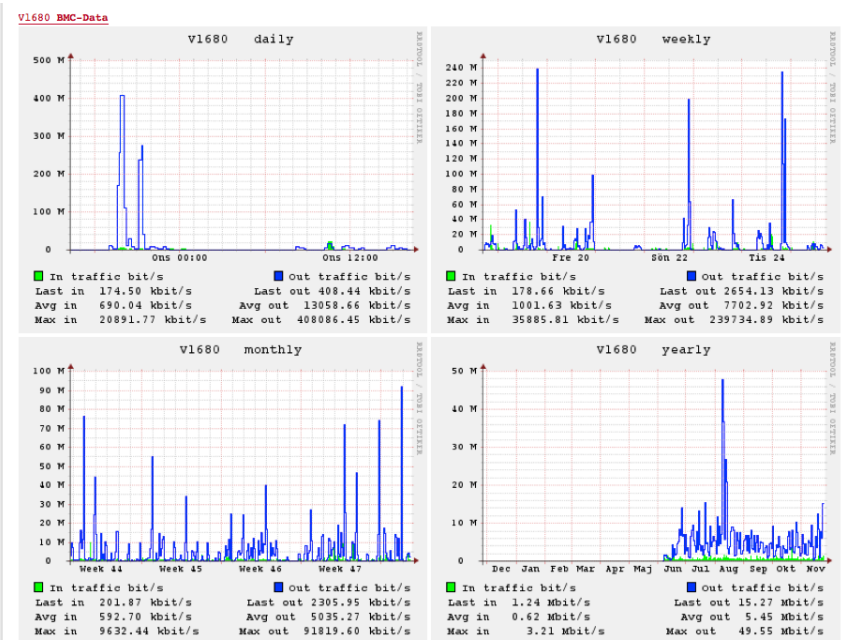
7 Other data collected and displayed

7.1 **Leasewatch** was collecting data from the DHCP-server regarding the dynamic DHCP-leases and sending warnings to responsible persons (looked up via RP-records) and writing rrdtool graphs. Currently not active.

7.2 **Arpsearch** There is a simple tool that is keeping a eye out for certain MAC-addresses and sends a message when they show up again.

7.3 **Arpwatch log search** Search the arpwatch log for events involving an IP- or MAC-addres.

7.4 **Linkstatus** Also rrdtool is used to plot bandwidth graphs for physical and virtual interfaces.



8 Thoughts for the future

- First of all, maybe some existing system exist that can do a lot of this stuff. What can Fortigate and BlueCat do?
- Run this or some other system all over the university network so that system administrators can have better control over the network. Make an API so that those who need it can access it. But I personally do not believe in some big revolutionary change but rather smaller step-by-step changes into something better.
- If this system should continue to run (better) then the data should be stored in some more serious database of some kind. At the same time set an API.
- *Arpwatch* (or the system formerly known as such) could report activity that does not comply with the current DHCP-server configuration.
- *Arpwatch* could store not only when the last activity was but also log when no activity is seen.
- *Arpwatch* could also consider not only when an MAC-address is new for the whole network but for a specific subnet or Vlan.
- *Porterwatch* could start logging events and storing when a port go up.
- Session information should be stored and saved.

- The cross connect and network socket information should be stored in a database authoritative rather than being imported and synced from Excel.
- It should be possible to lookup firewall rules affecting a certain IP. This could be done with data from Netreg where the router ACLs are published together with the IP as a key.

9 References and inspiration

Arpwatch by LBNL's Network Research Group
http://ee.lbl.gov/

GULP: A Unified Logging Architecture for Authentication Data by *Matt Selsky and Daniel Medina* - Columbia University
https://www.usenix.org/conference/lisa-05/gulp-unified-logging-architecture-authentication-data

Nagios by Ethan Galstad and others
https://www.nagios.org/

Observium by Adam Armstrong, Tom Laermans and Mike Stupalov
http://www.observium.org/

RRDtool by Tobias Oetiker, Oetiker+Partner AG
http://oss.oetiker.ch/rrdtool/

Zabbix by Zabbix Company
https://www.zabbix.com/