SDN meets transparent proxy

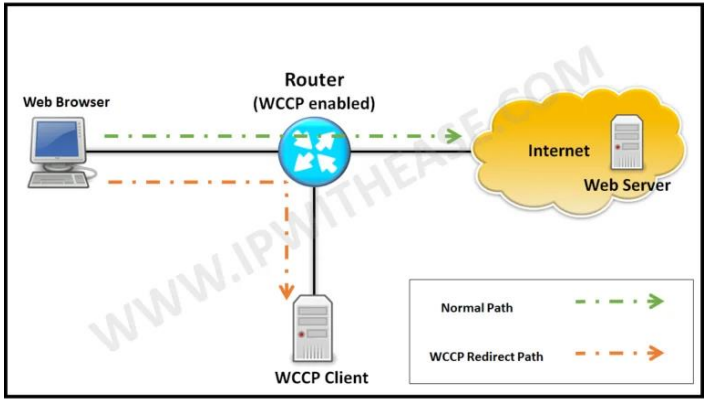
- Δημήτρης Καλογεράς, Phd,
- Μαρίνος Δημολιάνης, Phd
- ΕΠΙΣΕΥ/ICCS

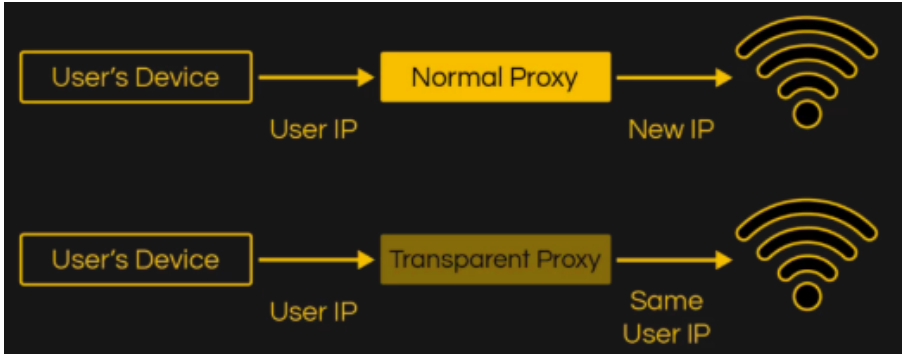
Agenda

1. Environment
2. The Problem
3. SDN
4. Detect
5. Act
6. Monitor

Environment: Content filtering for School Environment via HTTP Proxy

- Greek School Network
  - 14k schools operating ca. 2000
- 2 DC (Athens & Thessaloniki)
- Various Directory Enabled Services
- Transparent proxy as **redirection**
  - no user authentication
- authentication for proxy only via auto-proxy.. difficult to control this on WAN scale





Environment - Filtering

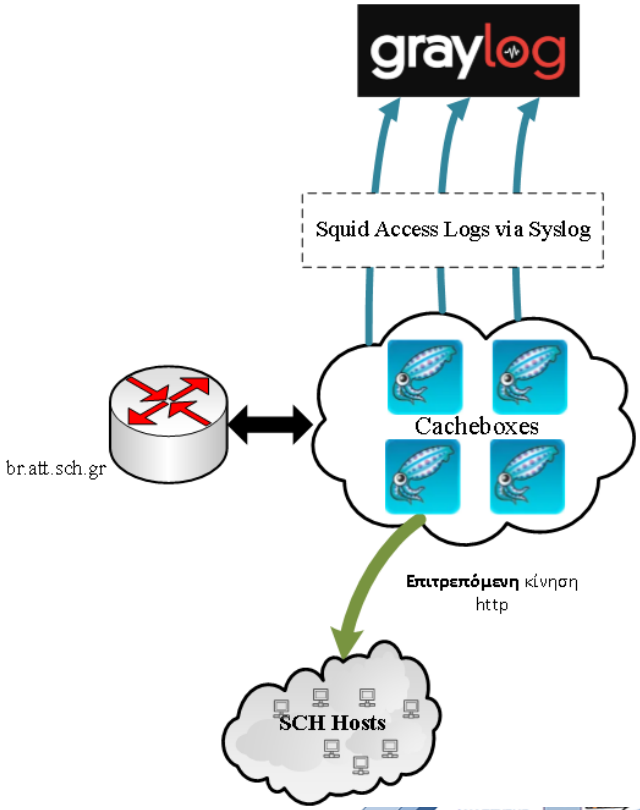
- HTTP and DNS based Content Filtering
  - DNS blocking via Response Policy Zones (RPZ) using public lists for crime, porn, bet, drugs and malware (DGA)
  - URL filtering using Squid proxy with ufdbGuard in transparent mode
    - x11 FreeBSD13 hardware boxes
    - 1x Docker-based on VM (Under testing)
- Proxy Farm control (insertion / removal of Proxies for redirection)
  - Static: ip route ..
- WCCP (dynamic insertion and withdrawal via heartbit)
  - Hardware assisted L2 traffic redirection to Squid proxies, crucial for large rates
  - UDP based Heartbeat control for High Availability for big farms

Problems Statement

- Microsoft (again) Windows Updates (MSU)
  - Big percentage of HTTP traffic, overwhelming proxy's resources
- **How to exclude traffic related to Microsoft Windows Updates?**
  - WCCP allows traffic exclusion by specifying destination IP addresses but
  - Microsoft content is served by IP addresses that are
    - not predefined
    - changing constantly (using multiple CDNs)
- **How to identify IP addresses serving Microsoft Content?** **Problem #1**
- **How to constantly update IP addresses serving Microsoft Content?** **Problem #2**

Problem 1 : How to identify IP addresses serving Microsoft Content?

- Leverage the actual proxied traffic to associate IP addresses corresponding to domains:
  - .*.microsoft.com
  - .*.windowsupdate.com
- How to collect such information?
  - Squid & Syslog ->
    Graylog (Elasticsearch & MongoDB)



Important

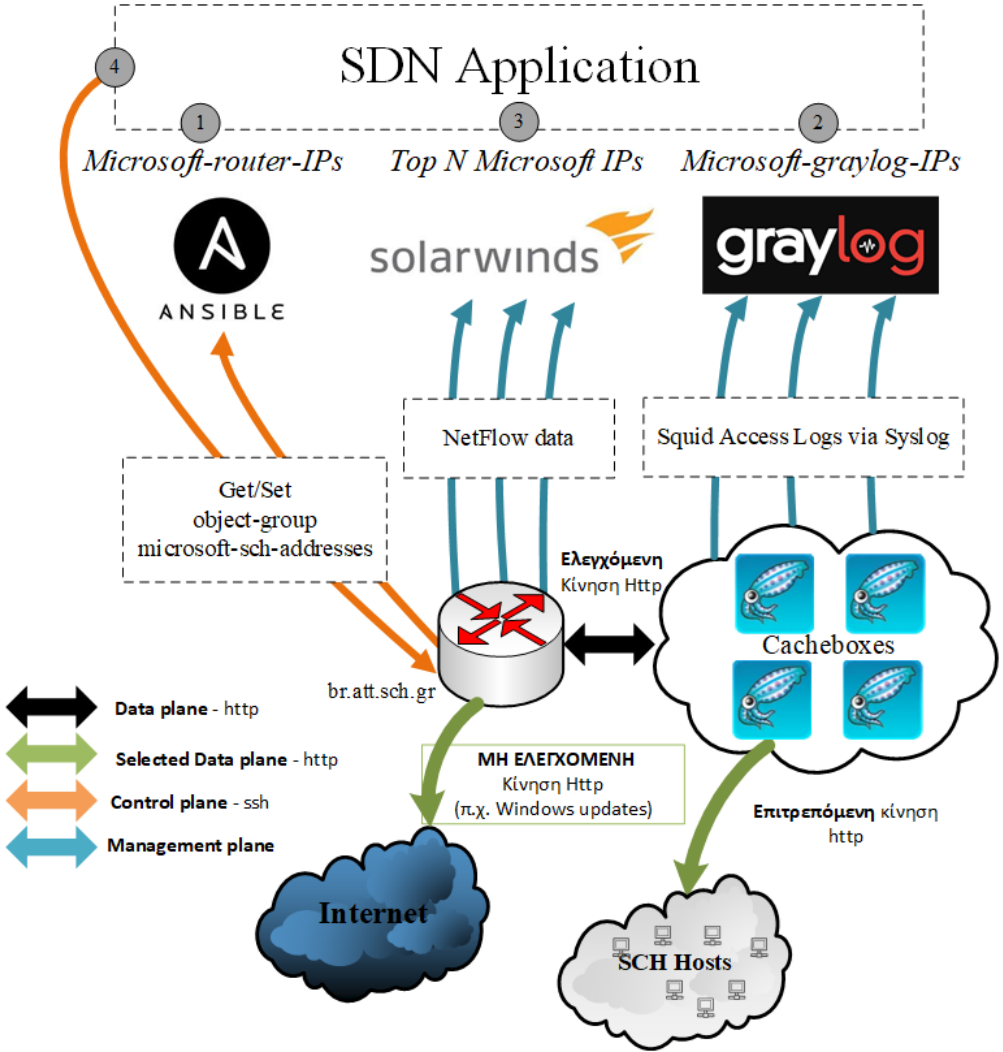
<u>The identified IP addresses are serving (at the time of log collection) Microsoft-related content and may change in the future. The only way to identify the Microsoft-related IP addresses is via the proxied traffic.</u>

How to constantly update IP addresses serving Microsoft Content

- Software Defined Network (SDN) solution
- Legacy way *‘Control Loop’*
    - Detect ←—+ (in the data plane: i.e. actual traffic)
    - Get : | the state of the router
    - Act | (in the configuration plane: i.e. via cli the ACL groups)
    - Monitor — + (via Netflow)

Objective : Keep MS downloads out of proxy infra

- Important build a generic mechanism though


SDN Application
4
1
3
2
Microsoft-router-IPs
Top N Microsoft IPs
Microsoft-graylog-IPs
ANSIBLE
solarwinds
graylog
NetFlow data
Squid Access Logs via Syslog
Get/Set
object-group
microsoft-sch-addresses
Ελεγχόμενη
Κίνηση Http
Cacheboxes
br.att.sch.gr
Data plane - http
Selected Data plane - http
Control plane - ssh
Management plane
ΜΗ ΕΛΕΓΧΟΜΕΝΗ
Κίνηση Http
(π.χ. Windows updates)
Επιτρεπόμενη κίνηση
http
Internet
SCH Hosts

Day - 0

Methodology

Detect

- Retrieve IP addresses related to Microsoft Content from Graylog

Act

- Populate IP addresses ACL to the router (Cisco) to be excluded
  - Bypass ACL (with object-groups) for WCCPv2 in the router
  - Ansible for templated configuration and object-group propagation
  - **Side Effect problem**: ACL keeps increasing !!!

Result

- HTTP traffic from/to these IP addresses is excluded from the proxies

Day - 1

Problem

Traffic destined to excluded IP addresses is not proxied -> not visible Squid logs

Methodology

Get (previous) state

- Retrieve excluded IP addresses from the router (Cisco)

Detect

- Retrieve IP addresses related to Microsoft Content from Graylog
- Combine previous excluded IP addresses (from the router) with the newly discovered (from Graylog)
- Retrieve the top-n destination addresses ordered by received traffic (according to NetFlow data, via Solarwinds API)
  - Remove IP addresses with few/no Microsoft-related HTTP traffic

Act

- Populate the **top-n** destination addresses to the router -> tweakable ACL entries

Periodical execution of the pipeline every h hours.

Conclusion & Next Steps

Conclusion

Reduction of traffic passing through Proxy Farms while serving the same number of end users

Next Steps

- **Aggregation of IP addresses -> reduce ACL entries**
- **Extend the mechanism for multiple domains**