Application of the IWF CAI URL list as a network filter - ***ClaraCAP***

David Freedman

Group Network Manager – Claranet

LINX67 – November 2009

clara.net

Why?

- Claranet traditionally known as ISP from 1996 onwards
- Re-focus in 1999 towards business customerbase / managed services “MSP”
- Fully paid up member of the IWF
- Had to come up with a firm position on CAI
  - Decision to implement filtering of CAI content
- Subscribed to the CAI URL list as a result

Hard choices to make

- Opt in or opt out?
- Business, Residential or all customer base?
- Started with opt in model, (ClaraCap1) a simple proxy farm that you could configure to use if you wanted to be protected from inadvertently accessing CAI content
- Surprisingly, nobody opted in, despite publicity targeted at residential users.

clara.net

So opt-out then?

- Version 2 was planned to :
  - Have technical ability to opt out
  - (almost all) *internet* customer bases covered
    - Reason for this is that if you could bypass system by *simply paying for a business internet account* then am sure you could imagine the reaction from the press
- Implemented 2008

ClaraCAP2 - Brief

- To download the IWF CAI URL list
  - On a regular basis, to ensure information freshness
  - To not modify the content of this list in any way other than to not accept URLs which do not conform to HTTP RFC or are on our own network (and can be dealt with ourselves)
  - To not allow the entire contents of the list to be viewed by human eyes
    - Developers would work with an artificial staging copy
    - System not allowed to store the list in memory, only a hashed representation of it
    - Even if operator could get hashtable, it would be pointless

ClaraCAP2 - Brief

- To be able to *see* HTTP requests from customers
  - Only where such requests are likely to be for CAI material
    - *Transparently* such as not to alert the user to this
    - Not to interfere with said requests unless they are for CAI content
    - Not to log, store or leak information about said requests
    - Not to be able to provide operators with any information about said requests

ClaraCAP2 - Brief

- To be able to *filter* these requests
  - In such a way as to disallow the request to pass before any content is retrieved.
    - *Transparently* such as not to alert the user to this
    - Without logging, storing or leaking information about filtered requests
    - Without allowing operators to access any information about filtered requests.
    - Without causing the user to experience any unnecessary pain.
    - In such a way so that *opted-out* users can bypass filtering

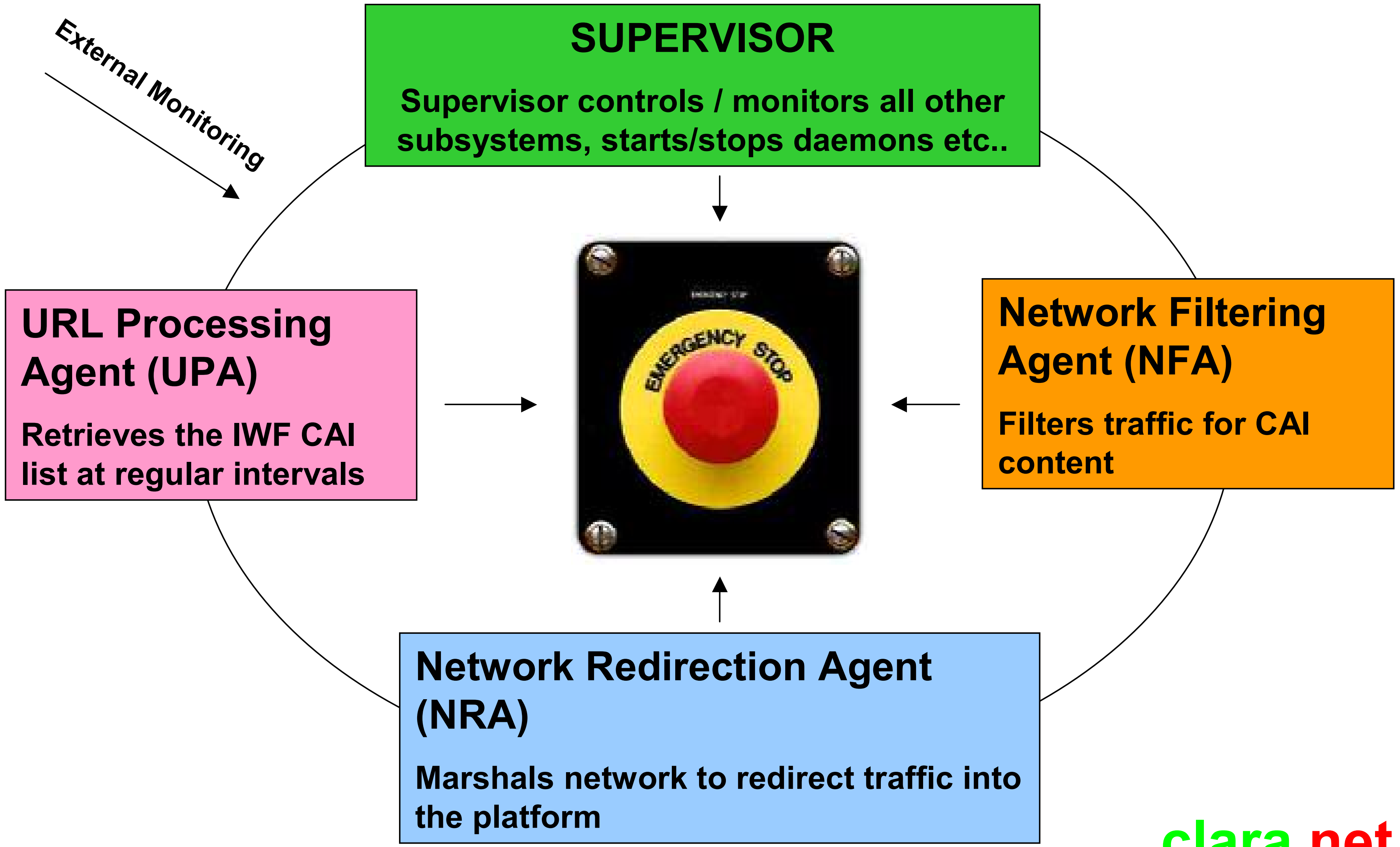
ClaraCAP2 - Brief

- To be wrapped up in operational security
  - To not allow the system developers or network operators access to the live (production) system.
  - To not allow all system operators access to the live (production system) – only a named subset meeting various internal criteria
  - To not allow system operators access to the system codebase for viewing or modification.

clara.net

ClaraCAP2 – Subsystem Design

Every subsystem has equal access to emergency stop



SUPERVISOR

Supervisor controls / monitors all other subsystems, starts/stops daemons etc..

URL Processing Agent (UPA)

Retrieves the IWF CAI list at regular intervals

Network Filtering Agent (NFA)

Filters traffic for CAI content

Network Redirection Agent (NRA)

Marshals network to redirect traffic into the platform

clara.net

ClaraCAP2 – Subsystem design

- panic() routine in each subsystem:
  - When it detects an abnormal condition
    - Informs the supervisor and asks it to panic() and shut everything down (which it does by pressing the emergency stop)
    - If supervisor times out, then we press emergency stop ourselves
    - If emergency stop times out then we terminate ourselves
  - Full stack traces written to panic logs but in operation **all CAI data is obscured**

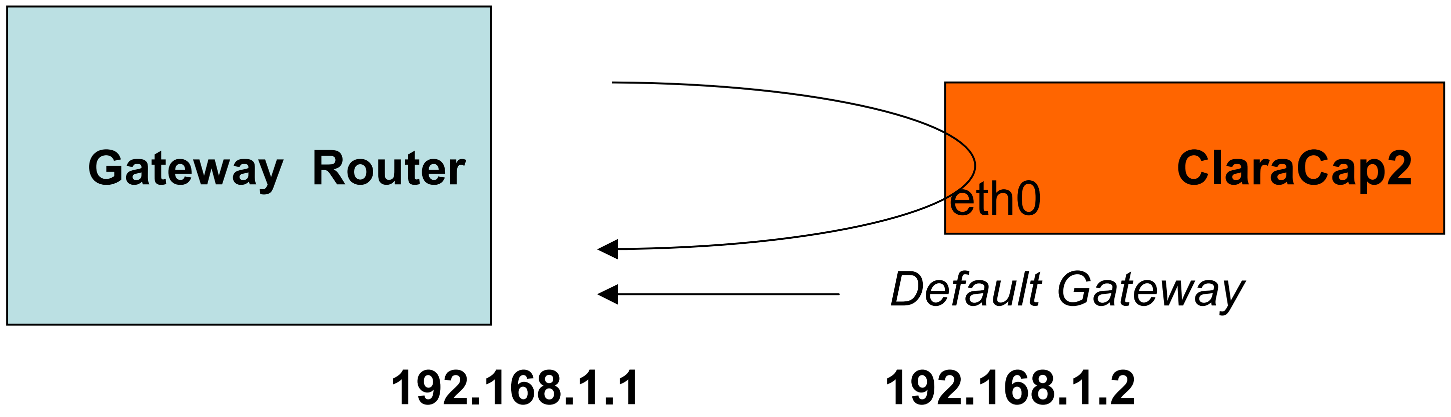
ClaraCAP2 – UPA Subsystem (URL Processing Agent)

- Retrieval URL, username and password stored on the box
- Hourly HTTPS GET to the Retrieval URL, HEAD taken first to confirm action required.
- Downloaded list stored directly in memory and syntax checked.
- All hostnames resolved into IP addresses asynchronously and using caching.
- IPs checked that they do not come from any of our (or our customers) netblocks (through expansive processing of our IRR AS MACRO) if they do, they are removed from the list and an urgent report is generated to our abuse team.
- IP List passed to NRA, hashed URLs to NFA, NFA must confirm it is ready before NRA instructed to redirect.
- Any failures cause subsystem panic()

clara.net

ClaraCAP2 – NRA Subsystem (Network Redirection)

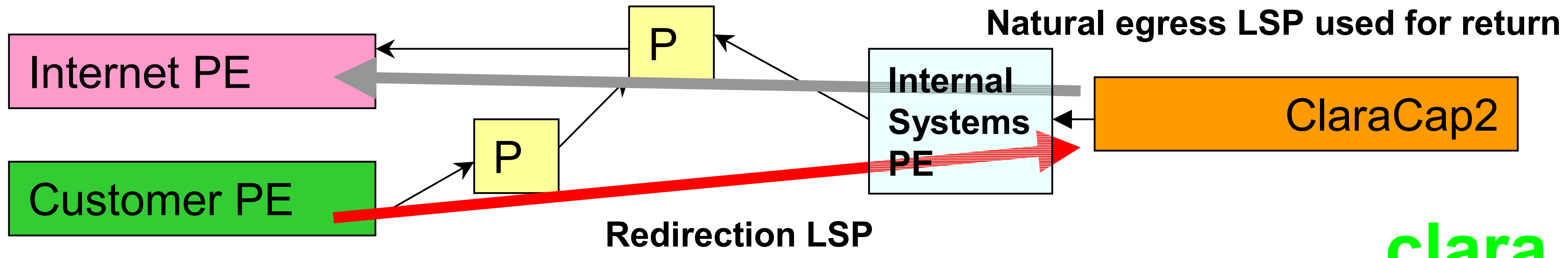
- Box is essentially a router
  - *echo 1 > /proc/sys/net/ipv4/ip_forward*
- But strangely on a stick



clara.net

ClaraCAP2 – NRA Subsystem (Network Redirection)

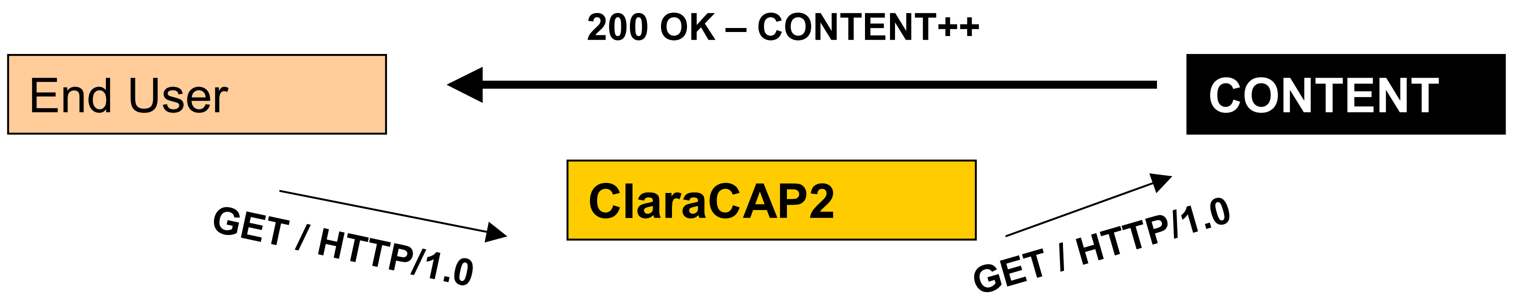
- "*Stick*" transfer /30 injected into our IGP (IS-IS in this case)
- MPLS in operation, IGP prefixes are therefore an MPLS *FEC* (target for MPLS LSP)
- Prefixes advertised by NRA into iBGP with Stick FEC as next-hop, causes customer PE to follow redirection LSP to stick network and into CAP
- Processed by NFA and if to pass, sent out default gateway to follow natural egress LSP



clara.net

ClaraCAP2 – NRA Subsystem (Network Redirection)

- Therefore CAP is only inline for HALF of a content request (the upstream half)
- Anticipate less traffic than if the reverse



- *“We can see what you are asking, but not what you are getting”*

clara.net

ClaraCAP2 – NRA Subsystem (Network Redirection)

- Prefixes injected by **custom BGPd** (based on openBGPd codebase) – NRA therefore maintains BGP sessions with ***all*** MPLS capable customer PE devices for the purposes of redirection
- Exposes NRA API to subsystems, redirection **only** allowed for **host routes**, each of which **sanity checked** before insertion, NRA only allowed to hold **finite** number of host routes at any one time (doubly enforced by **maxprefix** settings on PE routers)
- First sign of any trouble and subsystem panic() is called

ClaraCAP2 – NRA Subsystem (Network Redirection)

- Once traffic arrives at the box, passed to the NFA via kernel netlink – if NFA not available then routed back out whilst panic() called
- Once NFA has finished with packet, routed back out.
- If TTL would expire on this host then **incremented** by one before being forwarded out, this prevents system being shown in traceroutes.

ClaraCAP2 – NFA Subsystem (Network Filtering)

- NFA based on SNORT INLINE, accepts packets via NETLINK and processes or forwards them
- NFA only inspects TCP/80 packets, all else it passes
- HTTP request header in-line parsing performed
  - Request hash matched with NFA CAI hash list
- Positive match results in two actions
  - IP DROP
  - TCP RST sent to both parties
    - This undertaken to prevent user applications keeping sockets hanging around such that user feedback is provided promptly
  - **ABSOLUTELY NO LOGGING WHATSOEVER**

ClaraCAP2 – NFA Subsystem (Network Filtering)

- Technical Opt-out system implemented here
  - Source networks of all opted out customers stored in memory, read from customer database on startup
  - These are source routed around the NETLINK filter punt operation
  - Dynamic IP customers who wish to opt out have their addresses added and removed from the source list by push from radius/dhcp accounting systems (done by customer database flag)

ClaraCAP2 – System Monitoring

- I lied about not tampering with the list, it is indeed tampered with
  - We inject a test URL into it, pointing to a **primary** non-CAI image we generated ourselves on a web hosting account we have with a partner that is off the network
  - On the same account is a **secondary** non CAI image
- Monitoring system makes HTTP request for the primary and secondary images
  - If primary can not be retrieved but secondary can, the system works and is marked healthy
  - If primary can be retrieved and so can secondary, the system is not working, but must be offline, this is flagged for investigation
  - If neither primary nor secondary can be retrieved, the system is shut down as a precaution (emergency stop pressed remotely) and this is flagged for immediate investigation.
- Due to the web hosting provider having occasional outages we have now implemented a secondary web hosting provider and are using results to form a system health decision matrix.

clara.net

ClaraCAP2 – System Monitoring

- **Also, whilst we are talking about the list…**
- NFA internally monitors request count (per second)
- Sudden surge (+ delta) causes NFA panic()
- Designed to stop accidental insertion of popular, high traffic IPs into the NRA list
- If these are discovered then the IWF are contacted and informed but the system remains down until they advise.

Codebase

- Written in Perl
- Rather firmly integrated with our systems
- Parts of it can be made available, free on request
- Not covered by free software license nor freely redistributable however
- Anybody interested should contact me afterwards david.freedman@uk.clara.net,

Questions?

clara.net