Failures in a Hybrid Content Blocking System

Richard Clayton

University of Cambridge, Computer Laboratory, William Gates Building,
15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom

richard.clayton@cl.cam.ac.uk

Abstract. Three main methods of content blocking are used on the Internet: blocking routes to particular IP addresses, blocking specific URLs in a proxy cache or firewall, and providing invalid data for DNS lookups. The mechanisms have different accuracy/cost trade-offs. This paper examines a hybrid, two-stage system that redirects traffic that might need to be blocked to a proxy cache, which then takes the final decision. This promises an accurate system at a relatively low cost. A British ISP has deployed such a system to prevent access to child pornography. However, circumvention techniques can now be employed at both system stages to reduce effectiveness; there are risks from relying on DNS data supplied by the blocked sites; and unhappily, the system can be used as an *oracle* to determine what is being blocked. Experimental results show that it is straightforward to use the system to compile a list of illegal websites.

1 Introduction

There are a number of mechanisms for blocking Internet access to content. Barring particular IP addresses makes entire sites unavailable, but this can cause significant collateral damage when other websites share the same address. It is also possible to subvert the DNS so that websites cannot be located. Barring access to particular URLs is a more precise technology in that it can make specific parts of sites unavailable. However, it is much more expensive, requiring stateful inspection of packet contents within a firewall or the use of web proxies that interpose themselves between the requestor and the remote content.

In Britain there has been considerable interest in blocking indecent images of children (so-called "child pornography"). It has been illegal to "take" these images since 1978, illegal to "possess" them since 1988 and illegal to "make" them since 1994 [5]. The Internet Watch Foundation (IWF) operates a UK hotline for reporting illegal material found on the Internet. It collates the information it receives, and then informs the appropriate authorities. To avoid duplication of effort, the IWF maintains a database of URLs that have been inspected and keeps a record of when they led to illegal material. In particular, it became apparent to the IWF that although some countries took down illegal content promptly, some websites remained accessible for a considerable time.

BT is one of the largest UK ISPs, operating under brand names such as "BT Openworld", "BT Yahoo!", "BT Click" etc. In late 2003 they decided to create an innovative blocking system, internally dubbed “CleanFeed”.[1] Their aim was to prevent their Internet customers from accessing, either by accident or design, any of the illegal images of children listed in the IWF database. The existence of the system was leaked to the press [1] shortly before it became live in June 2004. The CleanFeed system is a hybrid design, incorporating both redirection of traffic and the use of web proxies. It is intended to be extremely precise in what it blocks, but at the same time to be low cost to build and operate.

This paper is arranged as follows: content blocking mechanisms are reviewed in more detail in Section 2 along with details of their worldwide deployment and previous studies of their effectiveness; the BT system is described in Section 3 and its effectiveness is considered in Section 4; the use of a hybrid system as an *oracle* to reveal which sites it is blocking is presented in Section 5 along with some actual results from the BT CleanFeed system.

2 Content Blocking Systems

2.1 Basic Mechanisms

There are three basic methods of blocking content available to ISPs and network operators. These are packet dropping (which operates at OSI layer 3), content filtering (operating at higher protocol layers), and DNS poisoning (to prevent any connection to the site being made at all).

Packet dropping systems are conceptually very simple. A list is created of the IP addresses of the websites to be blocked. Packets destined for these IP addresses are discarded and hence no connection can be made to the servers. The discarding mechanism can take note of the type of IP traffic, for example, it could just discard HTTP (`tcp/80`) packets and leave email alone.

The main problem with packet dropping is the collateral damage that it causes because *all* of the web content on the particular IP address will become inaccessible. This can be very significant. Edelman [4] obtained a list of all the `.org`, `.com` and `.net` domains and tried to resolve the conventional website address for each of them by prefixing the domain name with `www` and looking this up in the DNS. His paper shows that 87.3% of such sites share IP addresses with one or more other sites and 69.8% with 50 or more other sites. There is no reason to presuppose that content that might be suppressed is hosted in any special way, so his conclusion was that there is a significant risk of “overblocking” with schemes that suppress content by methods based solely on IP addresses.

DNS poisoning systems work by arranging that DNS lookups for the hostnames of blocked sites will fail to return the correct IP address. This solution also suffers from overblocking in that no content within the blocked domain remains available. Thus it would not be an appropriate solution for blocking content

[1] The official name for the project is the BT Anti-Child-Abuse Initiative.

hosted somewhere like geocities.com; blocking one site would also block about three million others. However, the overblocking differs from that identified by Edelman in that it does not extend to blocking other domains that are hosted on the same machine. There is also some "underblocking" in that a URL containing an IP address, rather than a hostname, would not be affected; because a browser would simply use the IP address and would not consult the DNS at all.

DNS poisoning can also affect other services, such as email. The blocking of right-wing and Nazi material mandated by the regional government in North-Rhine-Westphalia in Germany has been studied by Dornseif [3]. He found that the majority of local providers had opted for DNS poisoning but had made significant implementation errors. Although www.stormfront.org was (correctly) blocked by all of the ISPs he checked, only 15 of 27 ISPs (56%) also blocked stormfront.org as they should have done, and he believes that all but 4 of them only blocked it accidentally. Further, just 12 of 27 ISPs (44%) permitted access to kids.stormfront.org, which was not subject to a blocking order. Email should not have been blocked at all, but nevertheless 16 of 27 ISPs (59%) caused it to fail for some domains; and in the case of postmaster@www.stormfront.org, every one of the ISPs studied were (incorrectly) blocking email.

Content filtering systems will not only block entire websites but can also be used to block very specific items, such as a particular web page or even a single image. They determine that the URL being accessed is one of those to be blocked and then ensure that the corresponding content is not made available. This type of system is extremely accurate in blocking exactly what is on the list of URLs, no more, no less, and hence there should be no overblocking – provided, of course, that the list of URLs was correct in the first place.

Quite clearly, web proxies are ineffective at blocking content if their usage is optional. Hence it must be arranged that all customer traffic passes through the proxy, leading to a considerable expense in providing equipment that can handle the load. Also, to prevent a single point of failure, the equipment must be replicated, which considerably increases the cost. The bottom line for most ISPs considering blocking systems is that although content filtering is the most precise method, it is also far more expensive than the alternatives.

2.2 Existing Content Blocking Schemes

A number of content blocking schemes are known to have been deployed in various countries [13]. In China the current method appears to be a firewall scheme that resets connections [10]. Saudi Arabia operates a web proxy system with a generic list of banned sites, from a filtering software provider, augmented by citizen reported URLs [7]. In Norway, the child pornography blocking system introduced in October 2004 by Telenor and KRIPOS, the Norwegian National Criminal Investigation Service, serves up a special replacement web page "containing information about the filter, as well as a link to KRIPOS" [11].

In Pennsylvania USA, a state statute requiring the blocking of sites adjudged to contain child pornography was struck down as unconstitutional in September

2004. The evidence presented to the court was that ISPs had, for cost reasons, been implementing blocking by means of packet dropping and DNS poisoning. Careful reading of the court's decision [12] shows that the resulting overblocking was by no means the only relevant factor; no evidence had been presented to the court that the blocking had "reduced child exploitation or abuse"; and procedural mechanisms for requesting blocking amounted to "prior restraint", which is forbidden under the First Amendment to the US Constitution. However, the mechanisms actually deployed were significant, since the court determined that it was also "prior restraint" that future content at a website would in practice be suppressed, even though the abusive images of children had been removed.

3 Design of the CleanFeed System

The exact design of the BT CleanFeed system has not been published. This description is based on several separate accounts and although it is believed to be substantially correct, it may be inaccurate in some minor details.

The scheme is a hybrid, involving a first stage mechanism that resembles packet dropping, except that the packets are not discarded but are instead routed to a second stage content filtering system. The system is shown diagrammatically in Figure 1. The first stage examines all traffic flowing from customers (along the path labelled *a* in the figure). If the traffic is innocuous then it is sent along path *b* to its destination in the normal way. If the traffic is for a suspect site, parts of which may be blocked, then it is redirected along path *c* to the second stage filter. This first stage selection of traffic is based on the examination of the destination port number and IP address within the packets. The second stage filtering is implemented as a web proxy that understands HTTP requests. When the request is for an item in the IWF database a `404` (page unavailable) response is returned, but all other, innocuous, requests are relayed to the remote site along path *d* and the material returned to the customer in the reverse direction.

The IP addresses used by the first stage are obtained by working through all the entries in the IWF database and translating the hostname into an IP address in the normal way by making a DNS query. The results are amalgamated and used to modify the normal packet routing (controlled by BGP) within the customer-facing portion of the BT network (shaded in the diagram) so that the HTTP packets for these addresses will be routed to the web cache.

The second stage web proxy uses the URLs from the IWF database. Because there are concerns about keeping a human-readable form of the list on the server, it is held in what journalists have called an "encrypted form" (presumably as cryptographic hashes). The request is also "encrypted" (hashed) and a check for a match is then made. When there is no match, the proxy issues a request to the remote site in the usual way and then presents the response to the requestor. It is unclear, and not especially relevant to this discussion, whether the proxy also acts a cache, serving local versions of recently accessed material.

When compared with the generic solutions outlined in Section 2.1 and the systems deployed elsewhere in the world discussed in Section 2.2, the CleanFeed

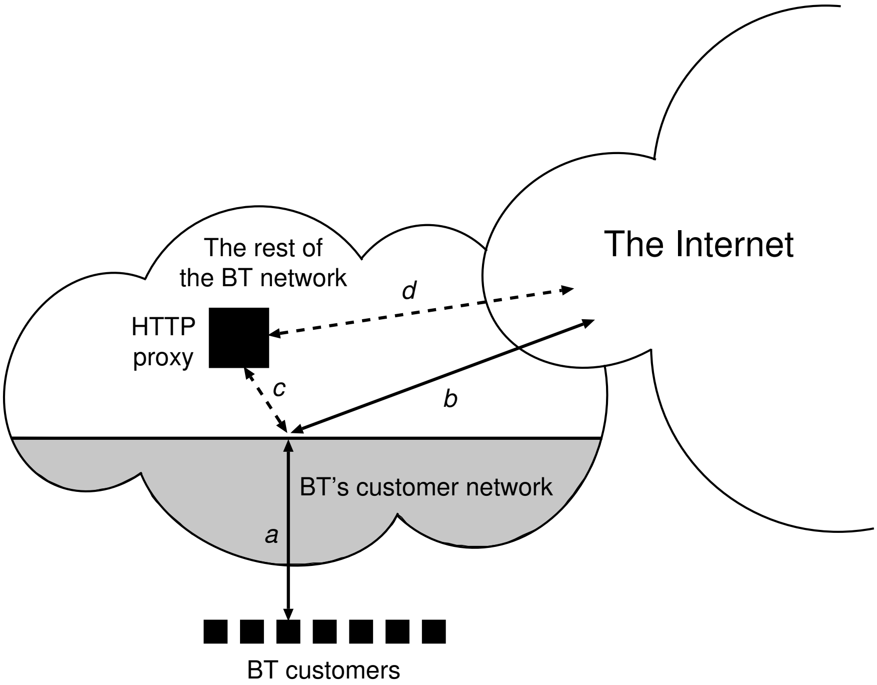

Fig. 1. The BT CleanFeed System

system has some significant advantages. Although its first stage uses the same approach as "packet dropping", it does not suffer from overblocking because the second stage web proxy can be as selective as necessary. However, the second stage can use low-cost equipment because it only needs to handle a small proportion of overall traffic. By avoiding DNS poisoning the designers can be sure that only web traffic will be affected and not other protocols such as email.

Therefore CleanFeed is, at first sight, an effective and precise method of blocking unacceptable content. However, there are a number of detailed implementation issues to address as soon as one assumes that content providers or content consumers might start to make serious attempts to get around it.

4 Circumvention of the CleanFeed System

4.1 Identifying the IWF and CleanFeed

If a content provider can identify that an access is being made by the IWF then they can provide information that is specific to that access. For example, they might provide innocuous images to the IWF and illegal images to everyone else. If they can do this successfully, the IWF will never blacklist their site and so it will not be blocked by CleanFeed. It is possible that some content providers already take this approach, since the IWF report that only 33% of hotline reports are substantiated as potentially illegal [6]. Identifying accesses by CleanFeed itself (in particular, components that do DNS lookups) also gives the content provider the opportunity for denial-of-service attacks as outlined in Section 4.3 below.

Table 1 summarises how a content provider might detect IWF/CleanFeed activity, along with the countermeasures that could negate these:

Table 1. Detecting content access by the IWF or the CleanFeed system

Content Provider Strategy	Countermeasure
Recognise the accessing IP address.	Access via web proxies.
Recognise source of DNS requests.	Use DNS proxies for name resolution. Ensure that CleanFeed access is random rather than a regular occurrence.
Anonymously report a sacrificial website to the IWF. Anyone who arrives to look at it must be the IWF or, later on, the police. Bar similar access in future.	Choose proxies and anonymous access systems likely to be used by genuine customers so that content provider will bar them as well.
Serve active content to run on viewer's machine and reveal their identity.	Disable Java, JavaScript, etc.
Serve cookies (from a central site) to tie visits to disparate sites together.	Refuse to return cookies (and/or clear cookie cache before browsing a new site).
Serve content with a request it be cached. Failing to refetch indicates a repeat visit.	Clear cache before browsing a new site.
Ensure unique URLs are used in advertising spam. A second access from a new IP address will be the IWF acting upon a report from the public.	Discard any obvious tracking information appended to URLs. Avoid starting visits in the "middle" of a website, but follow the links from the front page.

4.2 Evading CleanFeed

There are generic ways a content requestor (a customer) can avoid content blocking, such as tunnelling traffic via proxies that can access the content directly without any intervention. Dornseif [3] discusses this and a number of other techniques. For the hybrid CleanFeed system it is obviously effective to evade either of the two stages. However, CleanFeed's countermeasures can involve the first stage being less precise about traffic selection because the second stage will provide accurate blocking. Table 2 gives examples of some possible evasion strategies.

4.3 Attacking CleanFeed

Content providers could also actively attack the CleanFeed system, with a view to having it closed down. Example strategies and countermeasures are summarised in Table 3. Some build upon being able to identify CleanFeed system accesses and provide bogus information to them (see Table 1).

Table 2. Evading the CleanFeed system

Content Requestor Strategy	Countermeasure
Use a tunnelling technique, or proxy system, such as Tor, JAP, etc.	Also block the tunnels and the proxies (this is unlikely to be scaleable).
Use IP source routing to send traffic via routes that will evade the blocking.	Discard all source routed packets (often Best Practice anyway).
Encode requests (perhaps by using %xx escapes) so that they are not recognised.	Ensure URLs are put into a canonical form before they are checked.
Add specious characters to URLs (such as leading zeroes) to avoid recognition.	Ensure URLs are put into a canonical form before they are checked.
Provide specious HTTP/1.1 `Host:` details for an HTTP/1.0 site.	Check whether remote site acts upon `Host:` information.

Content Provider Strategy	Countermeasure
Move site to another IP address.	Regular updates of IP addresses.
Change port used for access (harder to track than address change, but may disrupt users as well as the blocking system).	Redirect other ports (paying careful attention if ports such as `tcp/53` (DNS) need to be intercepted).
Accept unusually formatted requests.	Extend canonicalisation to reflect what server accepts (which may be hard to do).

Table 3. Attacking the CleanFeed system

Content Provider Strategy	Countermeasure
Change location (IP address) of content very rapidly and often so that first stage routing "flaps".	Add addresses quickly, but remove slowly. No harm in sending extra traffic to the second stage unless it is overloaded.
Return specious DNS results referring to high traffic third-party websites. Hope to overwhelm the second stage web cache.	Avoid automating changes of IP address, and run sanity checks on returned values.
Return specious DNS results implicating BT customer or service machines, hoping thereby to create traffic loops.	Discard all results for external sites that claim to be inside the BT network.
Overload system by creating large numbers of addresses to block (eg by distributing content, perhaps by hijacking innocent machines to host the material [2]).	Unlikely to hit any limits in second stage web cache. In first stage, stop considering single addresses but redirect entire subnets. Provided cache can cope with traffic volume, no faults will be visible.

[2] The simplest way of providing content on large numbers of IP addresses is by means of a proxy network. In October 2003 Wired reported [9] that a Polish group were advertising "invisible bulletproof hosting" by exploiting a network of 450 000 end-user machines on which they had covertly planted their own software. Surreptitiously obtaining service from tens of thousands of machines is quite conceivable, so the Polish claim is not entirely outrageous, although without independent verification it cannot be seen as entirely trustworthy.

4.4 Blocking Legitimate Content

In a handful of special cases, the content provider can arrange for legitimate content to be blocked. Besides the inconvenience to those requiring access to this content, the effect is to bring the blocking system into disrepute and it must therefore be seen as an important attack.

Systems sometimes provide links based on IP addresses rather than by hostnames. This often occurs when linking to back-end database systems that serve query results or extracts from atlases. For example, a link to the Google cache of "the snapshot that we took of the page as we crawled the web" might be of the form `http://66.102.9.104/search?q=cache:FFKHU5mkjdEJ:www.cl.cam.ac.uk/users/rnc1/`. If so, then by ensuring that an illegal image at `http://www.example.com/search` was blocked by CleanFeed (using an anonymous hotline report) then the owner of the DNS for `www.example.com` can arrange for Google's cache to become inaccessible by serving `66.102.9.104` as a possible IP address for `www.example.com`.

The countermeasure is to ensure that whenever a DNS lookup yields new IP addresses they are checked for accuracy. However, if many IP address changes are being made by content providers in the hope of avoiding CleanFeed blocking altogether, it will be too expensive to manually check every change. Automated processes will be required for the testing that determines whether the content is the same but accessed via a different address. Unfortunately, an automated process cannot be relied upon to distinguish between an illegal website changing both content and IP address, and a spuriously supplied IP address. Hence, automation could lead to CleanFeed's users finding some legitimate sites blocked and this in turn would lead to a devaluing of the system's reputation.

5 Real-world Experiments

This paper has briefly discussed a number of attacks that might be made on the effectiveness or integrity of the CleanFeed system – and then explained how they might be countered. It would clearly be useful to determine which of the attacks are effective in practice and which are defeated either because the CleanFeed system already contains a countermeasure or because of some other aspect of the design that is not immediately apparent. However this is not possible, as will now be explained.

5.1 Legal Issues when Experimenting upon CleanFeed

Most experiments upon the CleanFeed system would require an attempt to access the sites containing the illegal material that the system is intended to block. If an evasion method was attempted and was successful in evading the blocking, then, under UK law, a serious criminal offence would be committed by fetching indecent images of children. Although there are statutory defences to inadvertent access, these could not apply to an explicit access attempt.

Experimenting with the techniques available to a content provider would involve working with the providers of illegal content, which would be ethically questionable, even if it was not directly a criminal offence. Even demonstrating that the IWF's access was easy to distinguish (a pre-requisite for some of the attack techniques) would involve submitting a false report and thereby wasting some of their analysts' time by causing them to examine websites unnecessarily, which is undesirable.

There is a method by which experimentation could be done, without these legal and ethical problems. If a test site, containing completely legal images, was added to the IWF database then it would be possible to perform all the necessary experiments on user and content provider strategies – and, as countermeasures were added, assess their effectiveness. However, permission to add such a site has been refused, so the only people running experiments will be the consumers or providers of illegal content and they are unlikely to report their results.

Nevertheless, it was possible to experimentally demonstrate that a user can exploit CleanFeed to construct lists of illegal websites. This is undesirable and unexpected, and should be taken into account when discussing the public policy issue of whether to encourage this method of content blocking.

5.2 Locating Blocked Websites Using the CleanFeed System

The CleanFeed system redirects traffic for particular IP addresses to a web proxy that then determines whether a particular URL should be blocked. It is possible to detect the first stage action, construct a list of redirected IP addresses, and to then determine which websites are located at those IP addresses – and hence use the system as an *oracle*[3] for locating illegal images.

The list of redirected IP addresses is created by a special scanning program. This sends out multiple TCP packets, each to a different address, with the destination port set to 80 and a TTL (time-to-live) value that is sufficient to reach the CleanFeed web proxy, but insufficient to reach the destination IP address (thus it will not unnecessarily trip intrusion detection systems at a remote site). If the IP address is not being redirected then the TTL will be decremented to zero by an intermediate router that will report this event via ICMP. If the IP address is being redirected then the packet will reach the web proxy. If the outgoing packet is sent with a SYN flag then the web proxy will respond with a packet containing SYN/ACK (the second stage of the TCP three-way handshake) and forging the IP address of the destination site. If the IP address is sent without a SYN flag then the proxy should respond with a packet with the RST flag set (because there is no valid connection).

The program was instructed to scan a /24 subnet (256 addresses) of a Russian web-hosting company (of ill-repute), with the results shown in Figure 2. Note that for this scan the SYN bit was set in the outgoing packets, when the SYN bit was absent the same pattern was visible but the RST packet was discarded by a local firewall!

[3] *oracle* is being used in the sense of Lowe [8] as a system that will accurately answer any number of questions posed to it without regard to the consequences.

```
17:54:27  Starting scan of [~~~.~~~.191.0] to [~~~.~~~.191.255] (TTL 8)
17:54:27  Scan: To [~~~.~~~.191.0]  : [166.49.168.13], ICMP
17:54:27  Scan: To [~~~.~~~.191.1]  : [166.49.168.5],  ICMP
17:54:27  Scan: To [~~~.~~~.191.2]  : [166.49.168.5],  ICMP
17:54:27  Scan: To [~~~.~~~.191.3]  : [166.49.168.5],  ICMP
17:54:27  Scan: To [~~~.~~~.191.4]  : [166.49.168.9],  ICMP
17:54:27  Scan: To [~~~.~~~.191.5]  : [166.49.168.9],  ICMP
17:54:27  Scan: To [~~~.~~~.191.6]  : [166.49.168.13], ICMP
17:54:27  Scan: To [~~~.~~~.191.7]  : [166.49.168.13], ICMP

... and similar responses until

17:54:28  Scan: To [~~~.~~~.191.39] : [166.49.168.1],  ICMP
17:54:28  Scan: To [~~~.~~~.191.40] : [~~~.~~~.191.40], SYN/ACK
17:54:28  Scan: To [~~~.~~~.191.41] : [166.49.168.13], ICMP
17:54:28  Scan: To [~~~.~~~.191.42] : [~~~.~~~.191.42], SYN/ACK
17:54:28  Scan: To [~~~.~~~.191.43] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.44] : [166.49.168.5],  ICMP
17:54:28  Scan: To [~~~.~~~.191.45] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.46] : [166.49.168.13], ICMP
17:54:28  Scan: To [~~~.~~~.191.47] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.48] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.49] : [~~~.~~~.191.49], SYN/ACK
17:54:28  Scan: To [~~~.~~~.191.50] : [~~~.~~~.191.50], SYN/ACK
17:54:28  Scan: To [~~~.~~~.191.51] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.52] : [166.49.168.5],  ICMP
17:54:28  Scan: To [~~~.~~~.191.53] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.54] : [166.49.168.5],  ICMP
17:54:28  Scan: To [~~~.~~~.191.55] : [~~~.~~~.191.55], SYN/ACK
17:54:28  Scan: To [~~~.~~~.191.56] : [166.49.168.1],  ICMP
17:54:28  Scan: To [~~~.~~~.191.57] : [166.49.168.5],  ICMP
17:54:28  Scan: To [~~~.~~~.191.58] : [166.49.168.1],  ICMP
17:54:28  Scan: To [~~~.~~~.191.59] : [166.49.168.1],  ICMP
17:54:28  Scan: To [~~~.~~~.191.60] : [166.49.168.13], ICMP
17:54:28  Scan: To [~~~.~~~.191.61] : [166.49.168.1],  ICMP
17:54:28  Scan: To [~~~.~~~.191.62] : [~~~.~~~.191.62], SYN/ACK
17:54:28  Scan: To [~~~.~~~.191.63] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.64] : [166.49.168.5],  ICMP
17:54:28  Scan: To [~~~.~~~.191.65] : [166.49.168.9],  ICMP
17:54:28  Scan: To [~~~.~~~.191.66] : [~~~.~~~.191.66], SYN/ACK
17:54:29  Scan: To [~~~.~~~.191.67] : [166.49.168.13], ICMP
17:54:29  Scan: To [~~~.~~~.191.68] : [166.49.168.13], ICMP
... etc
```

Fig. 2. Results of Scanning for IP Addresses Redirected by CleanFeed

These results (the high order octets of the IP addresses have been intentionally suppressed) show responses of either an ICMP packet (for TTL expired) from one of BT's routers in their `166.49.168/24` subnet, or a SYN/ACK packet, apparently from the remote site, but in reality from the CleanFeed web cache machine. The results clearly show that the CleanFeed system is intercepting traffic to a number of websites hosted at the Russian supplier. The full results show a total of seventeen IP addresses being redirected to the web cache.

Of course, knowing the IP address of a website does not allow one to view the content (unless it is using HTTP/1.0 or the server selects one main site to serve when just the IP address is present). However, reverse lookup directories exist that provide a mapping from IP address to web server name (they are constructed by resolving entries from the list of top level domain names). One such directory is sited at `whois.webhosting.info` and this was used to check out the IP addresses that CleanFeed was blocking.

Typical results (again there has been some intentional obfuscation) were:

```
~~~.~~~.191.40   lolitaportal.****
~~~.~~~.191.42   no websites recorded in the database
~~~.~~~.191.49   samayhamed.****
~~~.~~~.191.50   amateurs-world.****
                 anime-worlds.****
                 boys-top.****
                 cute-virgins.****
                 cyber-lolita.****
                 egoldeasy.****
                 ... and 27 more sites with similar names
```

and in total there were 91 websites on 9 of the 17 IP addresses. No websites were reported as using the other 8 IP addresses that were being blocked. This may be because the content has moved and the IWF have yet to update their information, or it may be because they were sites hosted in other top level domains, such as `.ru`, that the reverse lookup database does not currently record.

Checking the other IP addresses, not blocked by CleanFeed, showed a higher proportion of nil returns, but similar looking names. It is not possible to say whether these sites are innocuous or just not known to the IWF at present.

For the reasons explained above, *none* of these sites have been examined to determine what sort of content they actually contain, but it is fairly clear that if one was deliberately setting out to view illegal material then the CleanFeed system provides a mechanism that permits one to substantially reduce the effort of locating it. Further, since domain names can misrepresent the content (purveyors of pornography do not follow a truth-in-advertising code) it permits such a viewer to weed out superficially alluring website names and only select the ones that the IWF has already determined will contain illegal material.

Experiments showed that scans could be conducted at rates up to 98 addresses/second using a simple dialup connection. At this rate it would take 500 days to scan the entire 2^{32} address space – or, more realistically, 160 days to scan the 32% of the address space currently routable.[4] To scan just Russian IP addresses (and the IWF claim that 25% of all the websites they know of are located in Russia) then this is approximately 8.3 million addresses, which would take just under 24 hours. A suitable "BT Yahoo!" dialup account that is filtered by CleanFeed is available for free and the phone call will cost less than £15.

5.3 Countering the Oracle Attack

The oracle attack described in the previous section works by determining the path the packets take towards their destination. It is hard to counter in practice. The packets being sent by the end user can be made indistinguishable from normal TCP traffic – so they cannot just be discarded by a simple packet filtering system. The responses are either ICMP packets or SYN/ACK packets, again the latter must be permitted to pass, so discarding the former would not do anything especially useful.

If a web proxy is deployed in the network before the first stage at which a routing decision is made (which currently seems to be the case with the "BT Click" pay-as-you-go connectivity product) then the oracle attack fails (the web proxy treats all the packets the same, whether or not they will be candidates for redirection). However, this is an expensive fix, and BT have been removing compulsory (transparent) web caches from their products for marketing reasons.

The scanning attack is defeated if the first stage proxy does not redirect the packets to the web proxy unless their TTL setting is sufficient to reach the remote site. However, this would be complex to configure and would require specialised hardware, rather than standard routers running standard implementations of BGP. Even with this fix, it would almost certainly still be possible to distinguish web cache responses by examining the detail of what was returned.

An alternative approach is to make the scan less accurate. If the CleanFeed system redirected traffic destined for more IP addresses than the minimum necessary, then the scan results would contain even more innocuous websites than at present. It may be entirely practical to redirect /24 subnets rather than individual /32 addresses, the only question being whether or not there would be a substantial increase in traffic to the web caches.

Another way of reducing accuracy would be to make the first stage redirection less predictable by introducing a statistical element. If sites were sometimes blocked and sometimes not, then the scan would take longer to be sure of its results. However, this might not be a viable option with existing equipment and it is rather perverse to defend a blocking system against attack by arranging that sometimes it fails to operate.

The easiest way of dealing with the oracle attack would be to detect it occurring, most simply by examining logs at the web proxy, and then treating it as "abuse" and disconnecting the customer. It would probably take an attacker some time (and a number of terminated accounts) to determine how to reduce the activity sufficiently to avoid being detected.

[4] source: `http://www.completewhois.com/statistics/index.htm`

6 Conclusions

BT's CleanFeed was designed to be a low cost, but highly accurate, system for blocking Internet content. At first sight it is significant improvement upon existing schemes. However, CleanFeed derives its advantages from employing two separate stages, and this hybrid system is thereby made more fragile because circumvention of either stage, whether by the end user or by the content provider, will cause the blocking to fail.

This paper has described attacks on both stages of the CleanFeed system and set out various countermeasures to address them. Some attacks concern the minutiae of comparing URLs, while others address fundamentals of the system architecture. In particular, the CleanFeed system relies on data returned by the content provider, especially when doing DNS lookups. It also relies on the content provider returning the same data to everyone. All of this reliance upon the content providers' probity could well be entirely misplaced.

The CleanFeed design is intended to be extremely precise in what it blocks, but to keep costs under control this has been achieved by treating some traffic specially. This special treatment can be detected by end users and this means that the system can be used as an oracle to efficiently locate illegal websites. This runs counter to its high level policy objectives.

Although legal and ethical issues prevent most experimentation at present, the attacks are extremely practical and would be straightforward to implement. If CleanFeed is used in the future to block other material, which may be distasteful but is legal to view, then there will be no bar to anyone assessing its effectiveness. It must be expected that knowledge of how to circumvent the system (for all material) will then become widely known and countermeasures will become essential.

An important general conclusion to draw from the need for a manual element in many of the countermeasures is that the effectiveness of any blocking system, and the true cost of ensuring it continues to provide accurate results, cannot be properly assessed until it comes under serious assault. Thinking of these systems as "fit-and-forget" arrangements will be a guarantee of their long-term failure.

Postscript

A few days after this paper was presented at the PET Workshop, Brightview (a subsidiary of Invox plc) announced [2] that the oracle attack it describes was also effective against "WebMinder", their own two stage content filtering system, used by the UK ISPs that they operate. Their design is architecturally similar to that of CleanFeed, but they are employing Cisco's proprietary Web Cache Communication Protocol version 2 (WCCPv2) to redirect suspect traffic to a number of patched `squid` proxy servers.

In their announcement, Brightview also claimed that although their system had been vulnerable, they had now made the oracle attack "no longer effective". What they had done was to change stage one of the system to discard all packets with a TTL of less than 24. This means that the scanning program has to use higher TTLs; and hence both the web proxy and remote sites will receive the packets and return SYN/ACK responses – and, it was claimed, that would prevent the two sites from being distinguished.

It is true that the exact method of attack described above is defeated (and was achieved with just a one line change to the WCCPv2 configuration). It is also true that the fix is rather more elegant than just described in Section 5.3 which was envisaged to involve using different TTL limits for every possible destination. Nevertheless, as had been predicted, it remains straightforward to distinguish the web proxy from the remote site whose content it is filtering. A simple technique is to send the scans with a high TTL (such as 128)[5], to evade the countermeasure, and then examine the TTL in the returned packets.

Consider this scanning example of the /24 subnet to which the Russian sites listed above have now moved (with some other internal renumbering):

```
Scan: To [~~~.~~~.234.51] : [~~~.~~~.234.51], TTL=49 RST
Scan: To [~~~.~~~.234.52] : [~~~.~~~.234.52], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.53] : [~~~.~~~.234.53], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.54] : [~~~.~~~.234.54], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.55] : [~~~.~~~.234.55], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.56] : [~~~.~~~.234.56], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.57] : [~~~.~~~.234.57], TTL=59 SYN/ACK
Scan: To [~~~.~~~.234.58] : [~~~.~~~.234.58], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.59] : [~~~.~~~.234.59], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.60] : [~~~.~~~.234.60], TTL=49 RST
Scan: To [~~~.~~~.234.61] : [~~~.~~~.234.61], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.62] : [~~~.~~~.234.62], TTL=49 RST
Scan: To [~~~.~~~.234.63] : [~~~.~~~.234.63], TTL=59 SYN/ACK
Scan: To [~~~.~~~.234.68] : [~~~.~~~.234.68], TTL=49 RST
Scan: To [~~~.~~~.234.69] : [~~~.~~~.234.69], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.70] : [~~~.~~~.234.70], TTL=59 SYN/ACK
Scan: To [~~~.~~~.234.71] : [~~~.~~~.234.71], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.72] : [~~~.~~~.234.72], TTL=49 RST
Scan: To [~~~.~~~.234.73] : [~~~.~~~.234.73], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.74] : [~~~.~~~.234.74], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.75] : [~~~.~~~.234.75], TTL=49 SYN/ACK
Scan: To [~~~.~~~.234.78] : [~~~.~~~.234.78], TTL=49 RST
Scan: To [~~~.~~~.234.79] : [~~~.~~~.234.79], TTL=59 SYN/ACK
```

The results show RSTs from machines that are not running web servers (and there is no response where the IP address is unused). All the other IP addresses respond with SYN/ACK, but the TTL is 59 $(64-5)$ for the nearby WebMinder web proxy and 49 $(64-15)$ for the Russian sites that were ten hops further away. In practice the Russian sites returned a range of TTL values such as 45, 46, 47

[5] Setting a high TTL means that the packets will reach the hosting sites, which may detect a “port scan”; hence this attack is more “visible” than the original version.

(reflecting minor network connection differences) and 113, 238 (reflecting alternative operating system choices for the initial TTL values), but the web proxy value was constant and very different from any value returned by any real site.

Clearly, there are steps that Brightview could now take to obfuscate this latest hint, and an arms race could result as ever more complex methods are used to distinguish a server running `squid` in a UK service centre from machines running many different types of web server in other countries. However, it is a general principle that, in situations like this, hiding your true nature is impossible. So the best that can be hoped for is to make the oracle attack arbitrarily difficult rather than defeating it altogether.

Acknowledgments

This work was supported by the Cambridge MIT Institute (CMI) via the project: "The design and implementation of third-generation peer-to-peer systems".

References

1. Bright, M.: BT puts block on child porn sites. Observer, 6 June 2004. `http://observer.guardian.co.uk/uk_news/story/0,6903,1232422,00.html`
2. Brightview Internet Services Ltd: WebMinder, a configuration for restricting access to obscene sites identified by the Internet Watch Foundation. 9 Jun 2005, 21pp.
3. Dornseif, M.: Government mandated blocking of foreign Web content. In: von Knop, J., Haverkamp, W., Jessen, E. (eds.): Security, E-Learning, E-Services: Proceedings of the 17. DFN-Arbeitstagung über Kommunikationsnetze, Düsseldorf 2003, Lecture Notes in Informatics, ISSN 1617-5468, 617–648.
4. Edelman, B.: Web Sites Sharing IP Addresses: Prevalence and Significance. Berkman Center for Internet and Society at Harvard Law School, Feb 2003. `http://cyber.law.harvard.edu/people/edelman/ip-sharing/`
5. Her Majesty's Stationery Office: Protection of Children Act 1978.
6. Internet Watch Foundation: Annual Report 2003. 22 Mar 2004. `http://www.iwf.org.uk/documents/20050221_annual_report_2003.pdf`
7. King Abdulaziz City for Science and Technology: Local Content Filtering Procedure. Internet Services Unit, KACST, Riyadh, 2004. `http://www.isu.net.sa/saudi-internet/contenet-filtring/filtring-mechanism.htm`
8. Lowe, G.: An Attack on the Needham-Schroeder Public-Key Authentication Protocol. Information Processing Letters, **56(3)** (1995) 131–133.
9. McWilliams, B.: Cloaking Device Made for Spammers. Wired News, 9 Oct 2003. `http://www.wired.com/news/business/0,1367,60747,00.html`
10. OpenNet Initiative: Google Search & Cache Filtering Behind China's Great Firewall. Bulletin 006, OpenNet Initiative, 30 Aug 2004. `http://www.opennetinitiative.net/bulletins/006/`
11. Telenor Norge: Telenor and KRIPOS introduce Internet child pornography filter. Telenor Press Release, 21 Sep 2004.
12. US District Court for the Eastern District of Pennsylvania: CDT, ACLU, Plantagenet Inc v Pappert, Civil Action 03-5051, 10 Sep 2004.
13. Zittrain, J., Edelman, B.: Documentation of Internet Filtering Worldwide. Harvard Law School. 24 Oct 2003. `http://cyber.law.harvard.edu/filtering/`