CITI Technical Report 09-01

Clouseau Evaluation for Peer-to-Peer Transfer Operations

Charles Antonelli
Jim Rees
David Richter
{cja,rees,richterd}@umich.edu

ABSTRACT

We evaluate whether Clouseau, a commercial product from SafeMedia, Inc., is effective in discriminating between risky and non-risky P2P operations. We construct a testbed and assess the Clouseau's efficacy in interdicting risky content while passing non-risky content in both laboratory and office settings, using a wide variety of applications and protocols. We determine that Clouseau effectively blocks access to content via known P2P protocols communicating with networks identified by SafeMedia as containing risky content, but also prevents legitimate communications and does not block alternative protocols for accessing risky content. Consequently, Clouseau is not completely effective.

March 5, 2008

Center for Information Technology Integration
University of Michigan
535 W. William St., Suite 3100
Ann Arbor, MI 48103-4978

Clouseau Evaluation for Peer-to-Peer Transfer Operations

Charles Antonelli, Jim Rees, David Richter

Center for Information Technology Integration
School of Information
The University of Michigan

March 5, 2008

1 Introduction

Peer-to-peer (P2P) file sharing networks, implicated in the exchange of copyrighted material, are also critical for legitimate file transfers in support of academic research. We evaluate Clouseau[1], a commercial product from SafeMedia, Inc. that blocks risky[*] P2P operations, in an experimental test bed. The goal of this project is to assess whether Clouseau is effective in discriminating between risky and non-risky P2P operations.

We realize at the outset that it is unlikely that a purely technical solution will sort out the risky from the non-risky traffic. Copyrighted content being traded without permission is often indistinguishable from content being traded with permission of the copyright holder. Fair use transfers are indistinguishable from those that are not. Cryptography can render protocols and content opaque to any filtering device. Transfers that are legal in one jurisdiction may be illegal in another.

A technical solution must therefore suppress risky traffic based on some evidence that historically has been closely associated with risky activity. Such a solution is necessarily less than perfect, as the association of the evidence to the activity is less than perfect, with the result that sometimes non-risky traffic will be suppressed or risky traffic will be allowed to pass. Constant monitoring and adjustment of this association, to minimize the number of incorrect traffic classifications, must accompany any technical solution. This monitoring and adjustment must occur in both the short term, such as when a faculty member needs to transfer a particular file immediately to meet a proposal deadline, and over the long term, such as when peer-to-peer software vendors modify their protocols in an attempt to bypass detection, or when networks previously known for hosting risky traffic present evidence that they have ceased to do so.

Several technical solutions and vendors occupy this space. Three types of solutions were examined in a recent Common Solutions Group workshop held at Virginia Tech on January 9, 2008, where leading vendors of detection and suppression technologies were invited to present and discuss the architecture and implementation of their products.

---

[*] We define "risky" traffic as network data that represents content for which the user transferring the data seemingly holds no license or other right to share that content. "Non-risky" traffic is network data that represents content for which the user transferring the data seemingly holds a license to or has the right to share that content.

As summarized in the workshop observations:

> Audible Magic's CopySense technology can reliably identify only material that is registered with the vendor. Moreover, even modest encryption enables peer-to-peer traffic to bypass Audible Magic's detection.
>
> Red Lambda's cGrid technology detects traffic patterns rather than suppresses infringement. It requires considerable administrative expense and specific network architecture and management tools to translate identification of patterns into suppression of infringement.
>
> SafeMedia's Clouseau technology blocks any communications its vendor deems undesirable. Network operators cannot override this blocking locally, even if the vendor blocks important non-infringing communications or otherwise disrupts network operations and effectiveness.[2]

This contrasts the strengths and weaknesses of each approach in sifting the evidence: direct inspection of the network traffic, which is foiled by encryption; analysis of the patterns in the traffic, which depends on accurate analysis of the patterns; and Clouseau's approach.

The first two approaches suffer from an arms race. Encryption of network traffic will prevent direct inspection of it absent the encryption key. Varying the patterns will hamper traffic analysis; for example, network scanners such as nmap[3] have been successfully evading network intrusion detectors for years by adjusting the timing and content of the packets they send. Certainly the file sharing community will continue to participate in this arms race, as evidenced by the recent proposal to extend the BitTorrent tracker protocol to obfuscate the peers it returns, to "prevent internet service providers and other network administrators from blocking or disrupting BitTorrent traffic connections that span between the receiver of a tracker response and any peer IP-port appearing in that tracker response[4]."

Clouseau's approach, which is promoted by SafeMedia as not depending on inspection of network traffic or traffic patterns, thus presents an interesting alternative that merits closer study.

2 Testbed

The CLEPPTO test bed is shown in Figure 1. A Clouseau filter is installed between a network server and a NetGear ProSafe 24 port gigabit switch. A pair of client PCs running Windows XP SP2 and RedHat Linux 2.6.23.1-49.fc8 are connected to the switch. The CITI third-floor network, consisting of approximately twenty client hosts using a variety of operating systems and P2P clients, is also attached to the Clouseau through the switch. The testbed operates at 1 Gbps; some CITI hosts operate at 100 Mbps.

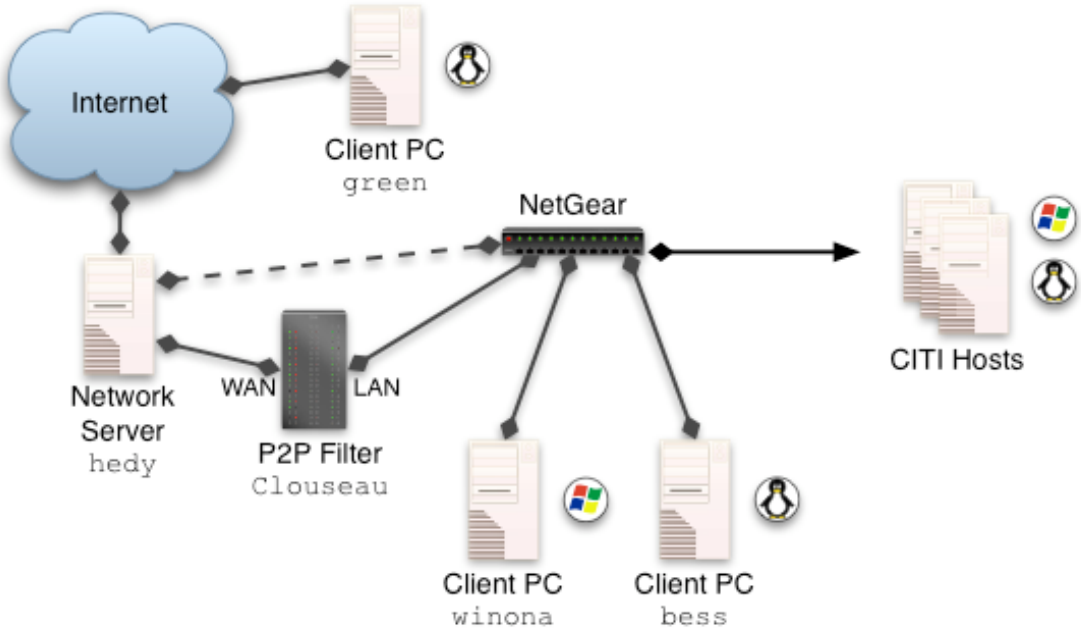

Figure 1

The network server runs RedHat Linux 2.6.23.1-49.fc8 and provides firewalled access to the Internet and packet monitoring services. Network monitoring software on the network server records traffic that successfully transits the LAN side as well as the WAN side of the Clouseau. The LAN side traffic is obtained via a connection to a monitor port configured on the NetGear switch; this connection is shown as a dashed line in Figure 1. We used tcpdump[5] to collect and store PCAP-format capture files, and Wireshark[6] to analyze the traffic.

A client PC running RedHat Linux 2.6.23.1-10.fc7 and sitting upstream of the network server is used for some of the tests.

We attached a laptop running Linux to the Clouseau serial port and used minicom[7] 2.1 to communicate with it. Once logged in to this serial console, it is possible to ask the Clouseau to display a line of information in real time for each packet it drops, or to dump this information in bulk to an ftp server on request.

We installed P2P applications on the client PCs, including BitTorrent and LimeWire, as well as applications that are not P2P software but that may exhibit network behavior similar to that of P2P traffic, such as NFS.

3 Testbed Experiments

A Clouseau 500 was installed in the CITI testbed as shown in Figure 1 on December 5, 2007. On December 14 it was replaced with a second Clouseau 500 at SafeMedia's request; this second unit remained installed until the end of the proposed testing period on December 21. The two units had slightly different enclosures but generally exhibited similar behavior. On January 4, 2008 it was replaced with a third Clouseau 500 at SafeMedia's request, containing a fix for a crashing problem we describe in Section 3.6.

During the test period we conducted two kinds of experiments. In the first, we exercised various P2P and non-P2P applications on the client PCs, attempting to download and upload both risky and non-risky traffic, and measured the performance of the testbed components. These experiments are described below.

The other kind of experiment consisted of CITI staff conducting routine activities using their CITI hosts, accessing services at CITI upstream of the Clouseau filter, on the University of Michigan's campus network, and on the Internet. These results are discussed in the next section.

Both kinds of experiments were conducted concurrently; we monitored network traffic at the network server and at the client PCs, and assessed bandwidth usage, network speed, and P2P client activity. Data gathered during both of these experiments was used to classify network transfer attempts into four classes:

Blocked Risky traffic (True Positive)	Unblocked Risky traffic (False Negative)
Blocked Non-risky traffic (False Positive)	Unblocked Non-risky traffic (True Negative)

3.1 Performance

Throughput and delay are two key performance measures for assessing the impact of any network filter. Accordingly, we used standard tools to obtain these measures with and without the Clouseau.

3.1.1 Setup

We installed the iperf[8] 2.0.2 client and server on hosts bess and green. We used the ping[9] utility to assess packet round-trip time, which we used as a measure of delay.

3.1.2 Test Procedure

On an otherwise idle testbed, we started the iperf server on green:

```
iperf –s
```

An the iperf client on bess:

```
iperf –c green
```

We measured TCP transfers, and used the tool's default TCP window sizes of 85 KB on the server and 16 KB on the client. Ten runs were obtained and discarded; the results of the next ten runs were averaged. We used the results obtained by the client.

We assessed round-trip delay using ping, for the smallest and largest possible single-packet payload sizes of 56 and 1472 bytes, respectively:

```
ping green
ping –s1472 green
```

Ten runs were obtained and discarded; the results of the next ten runs were averaged.

We then repeated the iperf tests in the opposite direction, reversing the roles of server and client.

Finally, all tests were repeated with the Clouseau taken out of the testbed, e.g., the NetGear switch connected directly to hedy.

3.1.3 Results

We report the results as observed by the client; the server results were commensurate. For round-trip time, the "server" is the ping destination. Results are shown in Table 1.

Clouseau	Server	Throughput	RTT size	RTT
In	green	927 Mbps	56	.486 ms
In	bess	938 Mbps	56	.486 ms
Out	green	941 Mbps	56	.406 ms
Out	bess	943 Mbps	56	.326 ms
In	green	-	1472	.688 ms
In	bess	-	1472	.694 ms
Out	green	-	1472	.548 ms
Out	bess	-	1472	.529 ms

Table 1

The Clouseau has a negligible effect on throughput, and introduces a 33%[†] increase in round-trip time for small packets and a 28% increase for large ones. However, this added delay would be negligible between most hosts on the Internet.

3.2 LimeWire

LimeWire[10] is one of many P2P clients that implements the Gnutella[11] protocol and is widely used to share risky content.

[†] Setting the green->bess RTT equal to the bess->green RTT of .406 still yields a 20% increase.

3.2.1 Setup

We installed LimeWire 4.14.10 on a Mac OS X 10.4.1 CITI host. It was not possible to access www.limewire.com to download the client, so the host was brought out from behind the Clouseau in order to download it.

3.2.2 Test Procedure

We launched LimeWire and attempted to search for the string "ubuntu", with the intent of locating and downloading a Linux distribution.

3.2.3 Results

LimeWire put up a popup window stating that "LimeWire is currently connecting to the network." The browser status bar displayed "Website found, waiting for reply," and eventually "Can't display web page." No search results were ever returned.

3.3 BitTorrent

BitTorrent[12] is a file-sharing protocol that distributes the cost of uploading the file to its downloaders by requiring that downloaders provide some upload services to other downloaders. BitTorrent is a popular P2P technology for sharing risky as well as non-risky content.

3.3.1 Setup

We installed BitTorrent 4.27.2 on a Mac OS X 10.4.1 CITI host.

3.3.2 Test Procedure

From behind the Clouseau, we attempted to find and download an Ubuntu Linux distribution using BitTorrent by visiting www.safetorrents.com.

In a separate test, we placed the Mac on a network that was not protected by the Clouseau, launched the BitTorrent application, searched for "ubuntu", and started a download. We then brought the Mac behind the Clouseau and observed further progress of the download.

3.3.3 Results

Visiting www.safetorrents.com allowed us to locate an Ubuntu distribution and download it successfully. On the other hand, once behind the Clouseau we were not able to resume a BitTorrent download that had previously been in progress – the download did not continue, and no uploads were started. The SafeMedia Release Notes point out that the Clouseau will block transfers from a "contaminated"[‡] network.

3.4 BitTorrent Tracker

We conducted a BitTorrent test using a University of Michigan server, our own (non-risky) content, and our own tracker and torrent files.

[‡] SafeMedia defines a "contaminated" network as one known to SafeMedia to contain illegal file sharing content.

3.4.1 Setup

We installed BitTorrent version 4.2.2 on a Linux server running within the University but outside the Clouseau protected subnet. We also installed BitTorrent version 4.4.1 on a Mac on the protected subnet.

3.4.2 Test Procedure

On the Linux server we started a tracker:

```
/usr/bin/bittorrent-tracker --port 6996 --dfile dstate
```

then made a torrent of an existing data file:

```
-rw-r--r--  1 cja umatlas 18012 Jan 11 12:27 test.torrent


maketorrent-console --data_dir ~ --target \
www/test.torrent http://host-dns-name:6996/announce \
path-to-file
```

and seeded the torrent:

```
bittorrent-console --ip host-dns-name --save_as \
path-to-file www/test.torrent
```

We then copied the torrent file to the Mac and attempted to fetch the data file via BitTorrent. We first did this with the Mac on the outside network to verify that the server was set up properly, then moved the Mac to the Clouseau-protected net.

3.4.3 Results

With the Mac on the outside net, the BitTorrent fetch succeeded, although somewhat slowly at around 17Kbps. With the Mac on the protected net, the transfer was blocked.

Following the procedure outlined in the release notes, we asked SafeMedia to unblock our torrent server. They responded within an hour, and said the server would be unblocked within three hours.

After this, the torrent was still blocked. Again following the release notes, we brought up a web server on the torrent server and hosted the torrent file on it using http over SSL (HTTPS):

```
./shttpd -d /afs/atlas.umich.edu/home/cja/www -p 6990 \
 -s /afs/atlas.umich.edu/home/cja/ssl/umfs02.web_server
```

Then we fetched the torrent file from the web server, and again tried to fetch the data file. This attempt failed. We then brought up an insecure web server:

```
./shttpd -d /afs/atlas.umich.edu/home/cja/www -p 6990
```

After re-fetching the torrent file, the file transfer succeeded, at the same rate it experienced on the outside net.

3.4.4 Discussion

The Clouseau apparently needs to see the torrent file being fetched from an approved web site before it will white-list that torrent. This makes it impossible to share torrent files in a secure way.

3.5 http

As we observed several web sites to be unreachable through the Clouseau, we investigated both the nature of the blockage and which sites caused this behavior.

3.5.1 Setup

We used Web browsers (Safari on Macintosh; Mozilla, Firefox, and Opera on Linux; and Chimera on OpenBSD) and command-line tools (nc[13], wget[14]) on various hosts on the LAN side of the Clouseau. We created a script on hedy to manage a pair of tcpdump captures on both its NetGear and Clouseau connections during a LAN-side browser session; this allowed us to observe differences in the filtered and unfiltered traffic. We also used nc and wget to generate appropriately modified HTTP requests.

3.5.2 Test Procedure

We attempted to view web pages using various URLs, some of which were contrived while others were accessed by CITI staff in the course of normal operations.

3.5.3 Results

The Clouseau passes the three-way TCP handshake, but drops all transmissions of the HTTP GET packet from the LAN side. Consequently, the client sees no data packets whatsoever.

On closer inspection, we found the Clouseau dropped HTTP GET packets that contained the header:

```
Host: thepiratebay.org
```

Capturing the request, modifying this header to

```
Host: thepiratebayx.org
```

and transmitting the modified request with nc succeeded in transferring the page; wget yields a similar result using its `–header` argument. Use of a tool such as WebScarab[15] automates much of this analysis.

The `Host:` header is optional in HTTP/1.0, and therefore a browser need not supply it; Chimera falls into this category. Comparing an original version of Chimera with one modified to include this optional header demonstrated that the absence of the header enables the browser to access “contaminated” networks.

It thus appears that the Clouseau is inspecting HTTP application layer headers inside packet bodies, and basing its decision to drop packets on what it finds there. During our investigation, we found the Clouseau blocked access to a surprising collection of web sites; see Section 4 for details.

3.6 bbftp

The bbftp[16] application is a file transfer tool optimized for bulk data transfer over wide-area networks, using multiple independent simultaneous data streams. It is commonly used in the scientific community for exchanging large experimental datasets.

3.6.1 Setup

We installed version 3.2.0 of bbftp (client) and bbftpd (server) on bess and on pogo, another CITI host outside the Clouseau network running Linux RHEL5. We installed a 600 MB test file (a Linux distribution .iso file) on both hosts.

3.6.2 Test Procedure

We started the bbftp server on bess:

```
bbftpd -b -m 10
```

and invoked the bbftp client on pogo:

```
bbftp -m -p 10 -e "get bigfile" -u user bess
```

We used ssh[17] to connect to these machines to start the software, with another ssh connection to monitor progress. We varied the number of parallel ftp streams in this test, up to a maximum of 10. We also reversed the roles of bess and pogo.

3.6.3 Results

With the first two Clouseaus evaluated, we found that requesting several parallel streams would, in a fairly short time, crash the Clouseau entirely. The machine would reboot itself, usually several times in succession, and it would eventually redisplay its login prompt. During this interval it would pass no traffic. It never required more than ten successive transfer attempts to trigger this condition. Nothing was output on the Clouseau's serial console during these events.

To observe this condition with slower traffic, we interposed a 100 Mbps hub between the Clouseau and its WAN connection. In this configuration, we were not able to cause the Clouseau to crash, although it slowed down ssh connections passing through it and exhibited noticeably irregular delays over them.

We notified SafeMedia of this condition with our first Clouseau and, while the second was also susceptible, the third model appears to have addressed this problem. With our current Clouseau it is not possible to force a crash with bbftp.

All three Clouseau models sometimes misidentify bbftp traffic as GnuNet[§] traffic, and block it.

3.7 tor

Tor[18] is a distributed overlay network designed to anonymize TCP-based applications like web browsing, secure shell, and instant messaging. Clients choose a path through the

---

[§] "GnuNet" is a P2P protocol that, like bbftp, uses SSL for encryption.

network and build a circuit in which each node in the path knows only its predecessor and successor nodes. Traffic flows through the network in fixed-size cells, which are unwrapped by a symmetric key at each node (like the layers of an onion) and relayed.

3.7.1 Setup

We downloaded a Tor & Privoxy & Vidalia bundle (version 0.1.2.18a) from torproject.org and installed it on a Mac, outside the Clouseau network, running Mac OS 10.4.11. We also installed the Torbutton Firefox extension from addons.mozilla.org.

3.7.2 Test Procedure

We started the tor proxy, connected Firefox to it, and verified via packet dumps that it was fetching web pages via the tor network. Then we timed how long it took to fetch several web sites: umich.edu, google.com, and torproject.org. We then moved the Mac onto the network protected by the Clouseau and repeated the timings.

3.7.3 Results

The Clouseau did not block the tor traffic but slowed it down considerably. The average time to load torproject.org went from 10 seconds outside the Clouseau network to 25 seconds inside. We examined the packet traces and discovered that Clouseau was blocking tor requests to remote servers on port 9030. The tor client then connected to a different set of servers on port 9001 and these connections were not blocked.

We then published a web page at http://www.citi.umich.edu/tor/server/, on a server outside the Clouseau test network, and attempted to connect to it from Firefox running inside the test network. Neither the server nor the client had any tor software installed. The Clouseau blocked connections to the server.

3.8 NFSv4

NFSv4[19] is a modern distributed filesystem with strong security. We use it extensively at CITI, with some users accessing their home directories over NFSv4.

3.8.1 Setup

For this test, we used bess as the client and mounted NFSv4 exports from the servers sparta (a NetApp filer) and whisper (a Linux server); bess was behind the Clouseau and the servers were not.

3.8.2 Test Procedure

We had bess mount sparta using three different security measures: first, without encryption; second, with Kerberos encryption (krb5), where protocol headers are transmitted in the clear; and third, with Kerberos encryption and privacy (krb5p), where NFS headers are also encrypted.

In each test, we had bess copy a large file (700MB) from the server. After running the tests with sparta, we repeated the tests using whisper as the server. Then, to verify that the difficulties did not stem from NFSv4 itself, all of the tests were repeated with bess outside of the Clouseau.

During testing, once the Clouseau had started blocking traffic, we generally restarted it before trying another test.

3.8.3 Results

Without encryption, while the copy would sometimes work correctly, performing several copies in a row or performing other concurrent command-line tasks such as listing or changing directories would cause the Clouseau to start blocking traffic. With krb5 encryption, the copy nearly always ended up blocked by the Clouseau. With krb5p encryption, interestingly, the copy nearly always worked without being blocked, while the transfer was noticeably slower; we are not sure of the connection.

Whenever the Clouseau blocked NFSv4 traffic, it was misidentified as GnuNet traffic. Please refer to Section 4 for more on using NFSv4 in the presence of the Clouseau.

All of these tests returned correct results when run with bess outside of the Clouseau.

3.9 Thunder/Gigaget

Xunlei[20] is a popular Windows-based Chinese P2P application in widespread use around the world. It supports multiple protocols and uses P2SP ("peer-to-server-peer") to increase download speeds by sharing content not only between client peers, but also between clients and fixed servers. Because it appears not to rely on the usual P2P protocols, Xunlei transfers can appear as normal web traffic.

Xunlei is entirely in Chinese. A modified version of it called Thunder also has English-language support. Gigaget is an internationalized version of Xunlei that is somewhat redesigned and distinct from Thunder.

3.9.1 Setup

We downloaded and installed Gigaget and Thunder (with its English-language package) on winona, running Windows XP SP2.

3.9.2 Test Procedure

Separately, we started each of the clients and attempted to search for and download potentially risky content.

3.9.3 Results

Thunder was more difficult to use because most of the application's options and searchable content was not in English. However, browsing through content was possible, and most searches returned pictures of things like popular movie posters or album covers, and so items could still be identified. We were able to access risky content in the presence of the Clouseau when using Thunder.

Interestingly, this was not the case with Gigaget. Despite that it appears to be very closely related to Thunder, upon starting Gigaget the Clouseau immediately started blocking inbound packets from a server in China and identified the traffic as Xunlei. The Gigaget client never made any progress and was not usable.

3.10 Zattoo

Zattoo[21] is an Internet television company that achieves good performance by relying on P2P techniques to share streaming content between nearby Zattoo users.

3.10.1 Setup

On winona, we downloaded the Zattoo version 3.0.8 beta for Windows and signed up for the University of Michigan's IPTV trial, which offers about 10 popular television channels to campus users.

3.10.2 Test Procedure

Our tests consisted of starting the client and letting television shows run. We watched for jitter and audio problems and switched channels periodically. For comparison, we also tested Zattoo outside of the Clouseau.

3.10.3 Results

The Clouseau did not appear to have any negative impact on the Zattoo client. No traffic was ever logged as blocked by the Clouseau and quality did not degrade.

4 CITI Staff Experience

The CITI staff experiment began on December 5, 2007, when the Clouseau was installed. Approximately 20 staff machines were either directly connected to the the NetGear switch as shown in Figure 1, or were connected over a wireless 802.11g Access Point which was itself connected to the NetGear switch.

CITI staff were asked to go about their daily business and report any false positives, i.e. legitimate packets the Clouseau was dropping, as well as any false negatives, i.e. P2P traffic the Clouseau seemed to let through.

4.1 True Negatives (unblocked non-risky traffic)

Here are some of the many web sites we were able to access from behind the Clouseau:

www.berkeley.edu
www.umn.edu
www.google.com
www.stonesoup.org
www.sol.de
www.salineschools.com
www.umflyers.org (web login required)
directory.umich.edu (CoSign[22] WebISO login required)
wolverineaccess.umich.edu (Two-Factor Authentication: CoSign plus token required)
mfile.umich.edu (with cached credentials)
www.piratesinfo.com
ctools.umich.edu
nationalcity.com
capitalone.com
www.chase.com

4.2 False Negatives (unblocked risky traffic)

We copied two different kinds of content from a Mac located behind the Clouseau to pogo. One kind was a small collection of MP3 files representing content for which we obtained permission for reproduction from the copyright holder, and the other consisted of licensed content (from a DVD of a motion picture) whose copyright had lapsed. We used the scp utility to copy the content through an encrypted tunnel and rcp to perform unencrypted transfers. The Clouseau did not block either of these copy operations. As SafeMedia has stated that they perform no content inspection, these were expected results.

4.3 True Positives (blocked risky traffic)

The Clouseau drops packets from all of the P2P applications we tested and prevents any communication with peer applications on "contaminated" networks on the WAN side of the Clouseau, except for Thunder[**].

Access was blocked to websites hosting common P2P programs for download (e.g., www.limewire.com), as was access to some websites that can be used to find risky content, such as thepiratebay.org.

4.4 False Positives (blocked non-risky traffic)

4.4.1 Web sites

We compiled a short list of inaccessible web sites:
en.wikipedia.org
www.bittorrent.com
www.securityfocus.com
http://scienceblogs.com/cognitivedaily/
www.askdavetaylor.com
www.alltheweb.com
legaltorrents.com
www.ynetnews.com
del.icio.us
albion.facebook.com

After contacting SafeMedia about some of these websites, some became accessible, probably via the Clouseau periodic update mechanism, but others remain blocked.

These apparently innocuous sites include www.ynetnews.com (an English-language Israeli news site) or del.icio.us (a popular social-bookmarking site).

Sometimes sites would fail to load because of something obscure like a banner ad being hosted from a "contaminated" domain (e.g., ads.morpheus.com), despite that none of the

---

[**] It is reasonable to assume that Clouseau can block Thunder in the same manner as it currently blocks other P2P protocols.

actual content was. The practice of routinely rotating banner ads obscured the nature of the intermittent page-load failures somewhat.

4.4.2 POP3S

The Clouseau appears to block access to Google Mail on POP3S port 995 intermittently; when this occurs, the Mac OS X Mail application remains stuck in the "sending password" state.

4.4.3 IMAPS

The Clouseau occasionally blocks access to mail servers on IMAPS port 993; mail-readers fail to load content. On these occasions, the Clouseau identifies the traffic as belonging to Retroshare, a P2P program that uses SSL. This happened to four different users during the aggregated CITI staff experiment.

4.4.4 HTTPS

The Clouseau occasionally blocks access to University of Michigan web-based email, which traffic is encrypted with SSL. Google Chat over SSL is also occasionally blocked; when this occurs, the chat client displays “We're experiencing technical difficulties that may prevent your chats from being sent.” Access to a satellite television company's support website was also blocked. On these occasions, the Clouseau identifies the traffic as belonging to Retroshare.

4.4.5 bbftp

The Clouseau periodically blocks bbftp traffic, which is encrypted with SSL. On these occasions, the Clouseau identifies the traffic as belonging to GnuNet.

4.5 Clouseau events

On four separate occasions during the testing period, our Clouseau crashed with the symptoms described in Section 3.6. This happened with our first two units. This appeared to be load-related, and the Clouseau seemed more susceptible to crashing if a large amount of traffic had recently passed through it. Shortly before it crashed, it also appeared to degrade its performance noticeably, with some connections seemingly unaffected while others were disproportionately degraded. All of these observations were subjective, however – it is difficult to see inside a black box.

After each such event we transferred CITI users back to the unfiltered network while we investigated. We then reconnected the users back to the filtered network. After the fourth event, we terminated the CITI staff experiment on December 13, 2007.

SafeMedia sent us a third Clouseau 500 on January 4, 2008. This version was installed, and after testing with bbftp, we determined the crashing problem had been fixed. We placed our users behind this Clouseau and performed a final round of user testing, ending on January 18, 2008.

4.6 Circumventing Clouseau

SafeMedia's claims regarding the Clouseau primarily center around it blocking P2P traffic, and to aid in that pursuit the Clouseau also restricts access to websites offering P2P clients for download. It should be noted, however, that cached content of those blocked websites can still be viewed using Google or the Wayback Machine (www.archive.org), as well as unblocked "mirrors" of restricted sites.

However, when it comes to finding and distributing risky content, there are a variety of alternatives to traditional P2P software. Successfully accessing risky content using these methods is possible despite the presence of a Clouseau[††]. These are viable alternatives to P2P for accessing the content that the Clouseau is ultimately aimed at blocking, and a technological solution claiming to interdict all risky content must block these avenues as well.

4.6.1 IRC

IRC (Internet Relay Chat) has some characteristics in common with P2P: it is distributed, searchable, and indexed; it is used to share risky content; it is "free"; and it has legitimate uses. IRC appears as a wide variety of online chatrooms ("channels") where users can also search for and download different types of content from automated "bots". There are many IRC servers, and they can link together so that a given channel is visible to users connected to any of a number of different servers.

To acquire specific content, a user can go to a website that indexes content in IRC channels (e.g., www.packetnews.com) and search for a movie or album; the site responds with a specific IRC server, a bot's name, and an item number. To download the content, the user then opens almost any modern chat client, signs on to the server, and sends a message to the given bot requesting the specific item; the downloading process can also be automated. Considerable content is available, and channels, bots, and servers shift frequently to avoid being shut down.

In testing, we were able to access risky content on IRC in the presence of the Clouseau.

4.6.2 USENET

USENET is a large, decentralized discussion system that also allows file transfers. USENET is divided into heavily categorized newsgroups, each of which is similar to a threaded web forum: users can view and post messages and files in the newsgroups using a newsreader client. A newsgroup is not limited to a given server; rather, different servers each host copies of the content of various newsgroups and users find servers that have the newsgroups in which they are interested.

As with IRC, a user can go to a website that indexes content in USENET newsgroups (e.g., binsearch.info) and search for content; the site responds with a list of posts from various newsgroups that likely have it. The user then finds a server hosting that

---

[††]These may not count as false negatives, depending on one's interpretation of SafeMedia's claims.

newsgroup, connects, and downloads the content directly. As with IRC, a lot of risky content is available; nevertheless, USENET also has a wide variety of legitimate uses.

In testing, we were able to access risky content on USENET in the presence of the Clouseau.

4.6.3 The web

There are many websites that host risky content without the aid of a P2P distribution system. Instead, they often rely on having multiple redundant, geographically distributed servers to serve content. A common approach is to provide tiered services, where rate-limited downloads are allowed for free and high-speed, unlimited downloads come for a subscription price of usually less than $10/month.

Some sites (e.g., www.rapidshare.com) advertise free backup services but allow clients to upload just about anything. Often the content is automatically indexed and searchable, and other clients can download anything stored on the site. Other sites are specifically geared for a particular type of content, such as freetvdown.blogspot.com, which offers a large selection of contemporary television episodes. Both of these sites generate income with advertisements and tiered download services.

While the Clouseau could easily be configured to block access to specific sites like these, relatively simple technological solutions present themselves. For instance, an unblocked, innocuous proxy service could be used to gain access to blocked content; coupled with SSL, even suspect searches for titles like "Spider-Man 3 download" would be encrypted and opaque to the Clouseau. Much more sophisticated proxy setups are regularly used to bypass the Great Firewall of China. It is conceivable, perhaps even likely, that circumvention measures like this would become common if the domain-based access-controls that the Clouseau employs became more widespread.

4.7 Living with Clouseau

On a simple day-to-day basis, our experience with the Clouseau was mixed. In many respects, the Clouseau worked as advertised: it blocked every common P2P client we tried and prevented us from downloading most of them with the Clouseau in place. Despite the results of Section 3.8, most routine NFSv4 operations appeared to succeed without incident.

However, in some cases it was difficult to attribute observed problems to the Clouseau. For instance, an attempt to clone a Git[23] repository over NFSv4 failed. It was not clear whether expired credentials, NFSv4, or Clouseau was responsible. After checking for "usual suspects" like expired security credentials or errors in NFSv4 itself, the problem remained unsolved. Even with the knowledge that the Clouseau was in place, it took some time before anyone suspected that it might have been the cause.

This type of difficulty with identifying intermittent problems came back time and again, precisely because they were difficult to reproduce. Since most of the time NFSv4 seemed to live harmoniously with the Clouseau, it did not immediately seem likely that any problems would arise. This pattern repeated with things like secure web traffic: most of the time attempts to access things like webmail or online bank accounts would succeed, but sometimes users would be blocked entirely. In fact, nearly every SSL-related

problem was intermittent, including IMAPS, POP3S, HTTPS, LDAPS, and bbftp. This behavior is probably due to the Clouseau's ongoing traffic analysis.

As touched on in Section 4.4.1, the issue of blocked websites is a source of confusion. Generally when a webpage fails to load, it is because of a slow server, network partition, bad cable, or missing browser plug-in. Sometimes it took a while before the Clouseau came under suspicion, either because the websites were utterly innocuous but were somehow identified as "contaminated," or because of something like an advertisement from a prohibited domain causing an otherwise-accessible page to fail to load.

5 Summary

The Clouseau performs some functions as advertised – it drops packets from all of the P2P applications we tested and prevents any communication with peer applications on "contaminated" networks on the WAN side of the Clouseau, except for Thunder. It does not interfere with network communications using a number of legitimate, non-P2P protocols. With respect to performance, the Clouseau has a negligible impact on throughput and round-trip time.

In this evaluation we have contrived some unlikely scenarios in an attempt to explore the boundaries at which Clouseau will block packets. We believe this is a legitimate exercise, as deployment problems will become apparent at these boundaries.

As Clouseau does not depend on content inspection, it is not vulnerable to being blinded by encryption, and it is reasonable to expect Clouseau to continue to perform as P2P networks migrate to encrypted traffic. Clouseau is dependent on periodic updates of its fingerprinting information, to allow it to interdict new protocols.

However, Clouseau suffers false positives:

- It prevents HTTP traffic to web sites SafeMedia deems inappropriate. SafeMedia did allow access via their update mechanism to some of the sites we reported as inaccessible during the test, but others remain blocked.
- It intermittently interferes with the NFSv4 protocol.
- It intermittently interferes with several types of SSL-based traffic.
- It drops any HTTP request that begins with "GET /tor/server/", even when the URL refers to a regular web page and not a tor proxy.

And false negatives:

- Clouseau does not block several viable alternatives to P2P for accessing risky content.

Finally, with respect to privacy, Clouseau appears to be inspecting application-level packet headers.

6 Conclusion

The goal of this project was to assess whether Clouseau is effective in discriminating between risky and non-risky P2P operations.

Clouseau is effective at blocking access to content using known P2P protocols communicating with networks identified by SafeMedia as containing risky content. In achieving this result, Clouseau also prevents some legitimate communications and does not block several alternative methods for accessing risky content, as summarized in Section 5. Consequently, Clouseau is not completely effective at discriminating between risky and non-risky P2P operations.

It is therefore vital for any institution contemplating the use of Clouseau to be able to adjust the way the evidence is interpreted. SafeMedia provides a procedure whereby customers may ask that a particular BitTorrent publisher be unblocked. This procedure has several problems. First, it requires a correct diagnosis of the problem. Packet blocking by Clouseau is nearly indistinguishable from other kinds of network outages such as bad interfaces, switches, or cables, routing problems, network congestion, and other sources of packet loss in the Internet. Second, it requires manual intervention and can take up to three hours after the request has been made. Finally, while the user can request BitTorrent unblocking, the filtering device ultimately remains under the control of SafeMedia.

In those environments where suppressing risky traffic is of paramount importance, and where suffering some amount of "collateral damage" in the form of false positives and negatives and unblocking delays can be tolerated, Clouseau can be an effective solution. In other environments, including our own, where false positives block legitimate access to resources and false negatives can expose users to risk, Clouseau is less effective. Organizations wishing to use this – or any other – technology will thus need to decide for themselves where in the spectrum between universal risk and universal access they wish to be.

References

[1] http://www.safemediacorp.com/products/

[2] http://www.stonesoup.org/docs/copyright-technology.pdf

[3] http://insecure.org/

[4] http://bittorrent.org/beps/bep_0008.html

[5] http://www.tcpdump.org/

[6] http://www.wireshark.org/

[7] http://nixbit.com/cat/communications/telephony/minicom/

[8] http://dast.nlanr.net/Projects/Iperf/

[9] http://ftp.arl.army.mil/~mike/ping.html

[10] http://www.limewire.com/

[11] http://en.wikipedia.org/wiki/Gnutella

[12] http://citeseer.ist.psu.edu/cohen03incentives.html

[13] http://www.vulnwatch.org/netcat/

[14] http://www.gnu.org/software/wget/

[15] http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project

[16] http://doc.in2p3.fr/bbftp/index.html

[17] http://www.openssh.org/

[18] http://www.torproject.org/svn/trunk/doc/design-paper/tor-design.pdf

[19] http://www.citi.umich.edu/projects/nfsv4/

[20] http://blog.scuforum.net/

[21] http://www.zattoo.com/

[22] http://www.umich.edu/~umweb/software/cosign/

[23] http://git.or.cz/

SafeMedia's Response to the March 5th Report, Prepared by the Center for Information Technology Integration at the University of Michigan.

"Clouseau Evaluation for Peer-to-Peer Transfer Operations"

SafeMedia appreciates the in-depth testing analysis the University of Michigan conducted and offers the following observations.

Scope of Cousteau's operations:

The goal of the project as defined by the University of Michigan was to assess whether Clouseau is effective in discriminating between risky and non-risky P2P operations.

The scope of SafeMedia's "Clouseau" P2P active denial system is firmly founded within a deeper scientific foundation. SafeMedia actively detects and prohibits P2P protocols, which cause "**Inadvertent file sharing**". For perspicuity, "Inadvertent file sharing" will automatically cause legal exposures, increases security threats to individual and network users and waste's network resources and bandwidth.

Based on this interpretation, Clouseau successfully achieved the tasks of protecting a user's network. But, other factors play into the real functionality and protection topic.

1. We uniquely deny only P2P protocols, which are used as **digital content delivery systems and create inadvertent file sharing .** That would automatically exclude all other P2P applications such as voice over IP, Ruckus, Lionshare, etc. Those P2P applications do not create inadvertent file sharing.
2. Applications like Email, FTP, and IM are not impacted by Clouseau.

We agree with your reports conclusion that Clouseau's P2P active denial system does exactly what it was intended to do which is: "Deny P2P connectivity for P2P clients that allow inadvertent files sharing".

We understand that some P2P clients claim they do not cause inadvertent file sharing and that the user can disable uploads. Unfortunately after exhaustive examination of their programs, the users cannot disable uploads if they expect to receive downloads. So without allowing sharing, the program becomes useless.

Bit Torrent testing:

1. Clouseau's release notes clearly outline that downloading bit Torrent files must be accomplished in the same session that the torrent was downloaded. Resuming a download is not allowed since it can inadvertently download "risky" torrent files. This was a design criteria for Clouseau and it appears your report validated its functionality.

2. Clouseau's design is to look at P2Pclient actions. It was never designed to stop command line actions. SafeMedia's entire focus was on the massive and unmanageable use of P2P clients. Admittedly, using command line functions for massive transfers are rare and would only provide a one-way file transfer. We chose to not include command line transfers because they have almost no impact in light of the volume of massive worldwide P2P traffic.

bbftp

1. bbftp does not follow the standard RFC in packet handling and acknowledge your awareness of it.
2. You tested 1GB transmission on a Clouseau configured for 500 Mb capacities. Once you deployed a Clouseau 1000 for the 1gb network, all issues of performance and failure stopped. Properly sizing Clouseau deployments involves several network topology considerations. Our technical team is well versed in sizing and can eliminate the under sizing that you noticed in your testing.

Tor

Thank you for discovering this deficiency. Release 1.0.34 corrected this problem.

NFSv4

Thank you for pointing out this deficiency. Release 1.0.35 corrected this problem.

Thunder/Gigaget

1. Thunder: release 1.033 corrected this problem and now Thunder is blocked.
2. In reality Thunder differs from Gigaget and Xunlie in two separate areas: the handshake and the pier fetch.

False Negatives (unblocked risky traffic)

Clouseau looks only at P2P clients and will not stop an SCP utility to copy the content through an encrypted tunnel. The SCP utility (not P2P) was not in the design criteria of Clouseau.

POP3S

We have tested and documented via Pcap file that Clouseau will not block any mail applications. Clouseau is capable of blocking or not blocking, but occasional blocking is more than likely a local issue typically generated by issues with a local machine and its internal configuration. We do not see this as a problem that Clouseau can create.

HTTPS

Thank you for discovering this deficiency; it has been corrected in release 1.0.33

IRC and USENET

These applications were not part of Clouseau'sspecification. We do not block them.

Living with Clouseau and Summary:

1. Intermittent problems: We have spent a great deal of time testing SSL and NFSv4 and could not find any intermittent problems. One suggestion to help quantify the problem is to analyze the Clint machine Pcap files against the rejected packets logs from Clouseau. After completing this analysis, we found no false positive evidence from either SSL or NFSv4. Whenever an intermittent situation occurs and the packets already successfully passed through Clouseau, then the problem can be isolated to the client machine.
2. Although your testing efforts were extensive, **some of your testing included scenarios that Clouseau was never designed for.** Our solution was designed to work with only P2P networks that deliver digital content and create risky inadvertent file-sharing environments. If someone wishes to distribute "copyrighted" content, other alternatives like mail, ftp, irc, and command line exist which Clouseau will not block. Premeditated file pirates can find alternative distribution methods, but one fact is certain, putting Clouseau on a network will totally eliminated a user's ability to use P2P networks.
3. Clouseau does not block several viable alternatives to P2P for accessing risky content. Viable alternatives or P2P file-sharing work arounds do exist. Clouseau was designed to address the larger problem of file sharing. The facts are 99% of P2P infringe file transfers use P2P networks

New Features added to Clouseau after you completed your testing:

As of release 1.0.49 SafeMedia offers the following features:

1. The user can now selectively define which protocols should be blocked and which protocols will remain unblocked. Both block and unblocked protocols will be reported in the system log files.
2. An automatic redirection feature was added to allow for communicating dropped packets in real-time to either an email server, or web serverso the user can be alerted about an infringing activity that was performed from his machine. Additionally, and IVR or Voice activated file can be created in real time for the call centers alerting them that a user was engaged with an infringing activity, eliminating the need of speaking to a customer service representative.
3. Network administrators can change the frequency and timing of updates received, to meet the schedules of standard network operation parameters.
4. Network administrators can request an immediate update without having to wait for the scheduled update timeline.
5. Network administrator's can define specific date and time settings for regular shutdown to match regular shutdown schedules of the network.
6. Network administrators can defined an IP for remote FTP or Syslog to transfer Clouseau logs to a central location for further analyses.
7. For organizations that acknowledge a P2P problem exists but decide to take no active denial actions, the network adminstrator can setup Clouseau to monitor and report without dropping any packets. This would allow complete detection of over (650) P2P clients, encrypted or non-encrypted and then notification based on the organizations policy around "copyright" infringement.