Forensics, FOR

14th – 18th October 2013

ASSIGNMENT

Mayur Pant

1st December, 2013

Part 1: 8 pages

Part 2: 38 pages

Part 1

Contents

Total: 8 pages

Assignment Part 1 *Question 1*

Anatomy of an attack

Just as one would use a formal process for computer forensics, a professional intruder ("*hacker*") would use one for a computer attack.

Overview

Planning

- port scanning to see which ports are open - probing the server environment for version numbers
- researching those versions (or cross referencing against a vulnerability database) for exploits
- information harvesting: scouring the front-end for username information / password clues

Mobilization

- having assessed the environment and applicable exploits, to download the appropriate tools

Action

- social engineering: to contact the target and falsify a request[1]
- running the tools and their appropriate modules against the target ('Exploitation'[2])
- retrieving the results (e.g. gleamed credentials)
- entering the server and performing the required action
- monitoring ('snooping') dataflow to & from all open ports (`wireshark`, `ngrep`, `tcpflow` etc.)
- installing a trap door (so even if the vulnerability is patched the attacker can return)
- clearing log files to evade forensic detection
- 'touching' file access and modification times with false timestamps to evade forensic detection. Despite this, they will not be able to alter all logs — their logout times for example, or any overlying layers of loggers to which their user has no access (e.g. Apache server logs or 'undelete' data in unallocated sectors)

Assessment

- the actions that were used to compromise the server
- the value of the overtaken server
- the success of the cleanup operating to 'eradicate footprints' (evaluating whether the intrusion was detected)

[1] Infamous social engineer Kevin Mitnick feels it is easier to trick someone into giving a password than to spend the effort in cracking a system.
K. Mitnick, CSEPS Course Workbook (2004), p. 4, Mitnick Security Publishing.

[2] K. J. Jones , R. Bejtlich, C.W. Rose, Real Digital Forensics, Computer Security and Incident Response (2006), pp.82-83, Addison-Wesley

Planning

Bejtlich et al.[2] describe the planning phase as 'reconnaissance'.

In this phase of the attack the criminal has chosen a target.
Their intention is to evaluate their target for weakness.

1. The target is a machine
If so the attacker will utilize a variety of network analysis tools (most of which may be standard to a network administrator's toolkit, others which may be customized for intrusion scenarios).

The intention here is to survey the target for weakness; since security vulnerabilities occur every day, unless the server is constantly patched with the latest updates (and even if it is, the attacker may be exploiting an unrecognized bug, server misconfiguration or weak password), and the network administrator is 100% vigilant, there is likely to be some exploitable flaw.

We spoke of customized network analysis tools for intrusion scenarios; here we find 'all-in-one' tools[3] designed for penetration testing (such as `Metasploit`, `Wireshark` and `nmap` which accept additional modules). These tools collectively can fingerprint types of software environment and traffic flow, even being able to decrypt certain network protocols in realtime with specific modules.

Once these tools have probed the environment, the attacker will be able to compile a detailed list of the environment's software and open ports (sockets through which they can obtain a connection or launch an exploit).
The list will feature information such as the operating system version and detected software (such as a particular version of a running webserver, e.g Apache).

It is important to note that a significant amount of information can also be gathered through the front-end; the site may even advertise that it uses a particular framework (e.g. Wordpress, V.xx). A valuable supplement of front-end browsing is that it may provide a list of users who frequent and visibly post to the site (e.g. many forums display a list of users currently logged in). It is quite simple to retrieve such front-end visible user information by querying Google for particular site pages, searching for proper nouns or non-dictionary words that may represent usernames.

Having achieved a detailed list of the technical environment, the attacker must cross reference the gathered information against a security database[4] (e.g. BugTraq). For convenience, just as a virus scanner updates its database automatically to detect the latest threats, some penetration testing tools (such as `Metasploit`) update their databases to match the latest vulnerabilities against the detected environment.

2. If the target is a human
Additionally, potentially more valuable information may be obtainable from the site's administrator or users. In the case of a website the user may look up the domains administrator, and then contact the person (and any retrieved users), collecting additional information through the practise of social engineering[1]. As well as gathering information, one might coerce the target into performing operations conducive to the attack.

[3] a rated list of the best penetration testing tools rated by G. 'fydor' Lyon, the author of `nmap`, can be found at http://sectools.org
[4] 'fydor' also rates vulnerability lists at http://seclists.org

Mobilization

Having a detailed list of the environment cross referenced with a current list of vulnerabilities, the attacker now obtains the appropriate modules.

As we mentioned, an intelligent tool might acquire the database of vulnerabilities automatically. It may even provide associated references to the modules and their documentation.

If the modules are not-cross referenced and downloaded automatically, the attacker will want to review security lists[2] for the given software. Following the research, they will then download the appropriate tools (whether they be modules to the all-in-one tools, specific scripts, command instructions, injectable queries, or (compilable) standalone applications when no add-on is available) for the attack.

A professional attacker will not just download potential exploits and test them one by one 'on-the-fly'; it is prescient to download and research all of the appropriate tests that will be conducted, and launch them all at once in quick succession. The attacker that plans their attack will spend less time at the crime scene - for targets that employ a live security analysis team this is important. "*Quickly in, quickly out*" is the intention.

By this point the attacker has not just downloaded the modules – he or she has also constructed a plan of order in which the attacks will be launched, and structured any processes for leaving 'back-doors' and 'footprint clearing'.

Action

They now launch the attack plan (the 'exploitation'[2] phase). When they do, it will be best to perform the attack from a misrepresentative IP address (i.e. by using spoofing software or a proxy server in a different location, disguising the attacker's actual coordinates). The attacker will want no traceable link to the conduit launch platform; rather than renting a server ("with their own credit card"), they might utilize a botnet node (or previous target), which they might rent with an anonymous digital currency.

Provided one of the exploits worked, the attacker has now gained entry to the target. In the case of social engineering, the best case scenario is that the attacker has coerced a human representative of the target to provide them with an elevated permissions account, which the attacker is now connected with.

Depending on the intentions of the attacker (whether they be data-harvesting or defacing — see Figure A), the attacker now calls for 'reinforcement'[2]. They have an active shell; they will want to download their personal toolkit onto the target (e.g. by using `wget` or `scp` - standard userland applications that the target may have depending on their platform).

Having a local copy of the attacker's toolkit at the target may serve two functions:

1. The right tool may allow them to perform an additional exploit from their current shell to elevate their permissions further (i.e. an exploit of the webserver to gain a restricted shell is followed by an exploit of the operating system to gain an administrative shell);
2. Installing a 'backdoor', allowing them access to the machine in future without having to breach any vulnerability.

Point 1 will allow them to perform additional actions; whether it be installing a program (such as their backdoor), retrieving data from every home directory on the server, creating their own administrator account, performing a defacement or other (see figure A).

Point 2 allows them subsequent access to the server, even if the vulnerability they exploited is subsequently patched by the administrator (they might even patch it themselves in point 1, so as to mislead the forensics effort or prevent other attackers from infringing on their claim).

The attacker will want to 'consolidate'[2] their actions; in other words, having installed the backdoor they want to verify that the new portal is able to communicate successfully with their own server(s) (having opened any firewall ports if necessary). At this point, they might choose to 'pillage'[2] (performing any of their intended abuses then and there), or comeback later via their backdoor (especially likely if the exploited server is employed as a botnet or 'zero-day' distributed denial of service attack node).

Additionally the attacker will want to 'clean up any footprints' - that is, remove as much trace of their presence as possible. As Dr. Locard's Exchange principle states, "*the tool mark he leaves...whatever he touches..will serve as a silent witness against him. It is factual evidence*". So it would be wise for the criminal to 'untouch' any of their 'fingerprints' - that is, use a tool to reset any timestamps affected by their behaviour[5]. That said, they will find it very difficult to erase their final digital footprint - when they log out, the server logs should store this (unless their backdoor rewrites the logs after their departure; and even then the presence of their backdoor should be detectable). And it might be that some logs are abstracted a layer above what the intruder has access to (e.g. ISP logs that are unavailable to the webserver).

A professional attacker, like a professional forensic analyst, cannot afford to be negligent - the slightest trace could become vital case evidence.

Assessment

Depending on whether the attacker is using a repeat formal process for the attack, some of the actions chosen in the cleanup operation may be evaluated by an assessment.

The attacker will want to confirm they have made a silent exit. In doing so they might consider whether any trace evidence of their actions have been detected; Bejtlich et al.[2] describe that an attacker will often return via their backdoor "*when satisfied no one has discovered his presence*".

Finally, if the attacker is diligent, they might assess the value of the compromised server, and review the effectiveness of their attack (i.e. can the server now be utilized in subsequent attacks, and were the commands completely effective in harvesting the desired information).
Just as a police detective[6] or forensic analyst creates a formal report of a particular incident, the prudent criminal may review the steps taken to optimize the effectiveness of future breaches. They might even submit a report of the incident to a superior head (e.g. in the case of a botnet credit card harvesting operation, a syndicate leader).

[5] They might manually retrieve the timestamps of every file they plan to 'touch' immediately upon entering the environment; then run their tools; then reset these prior access and modification timestamps with a timestamp 'touch' tool.
[6] An NYPD incident form issued to the author (2013), http://users.ox.ac.uk/~kell3138/NYPD-2013-09-27.pdf

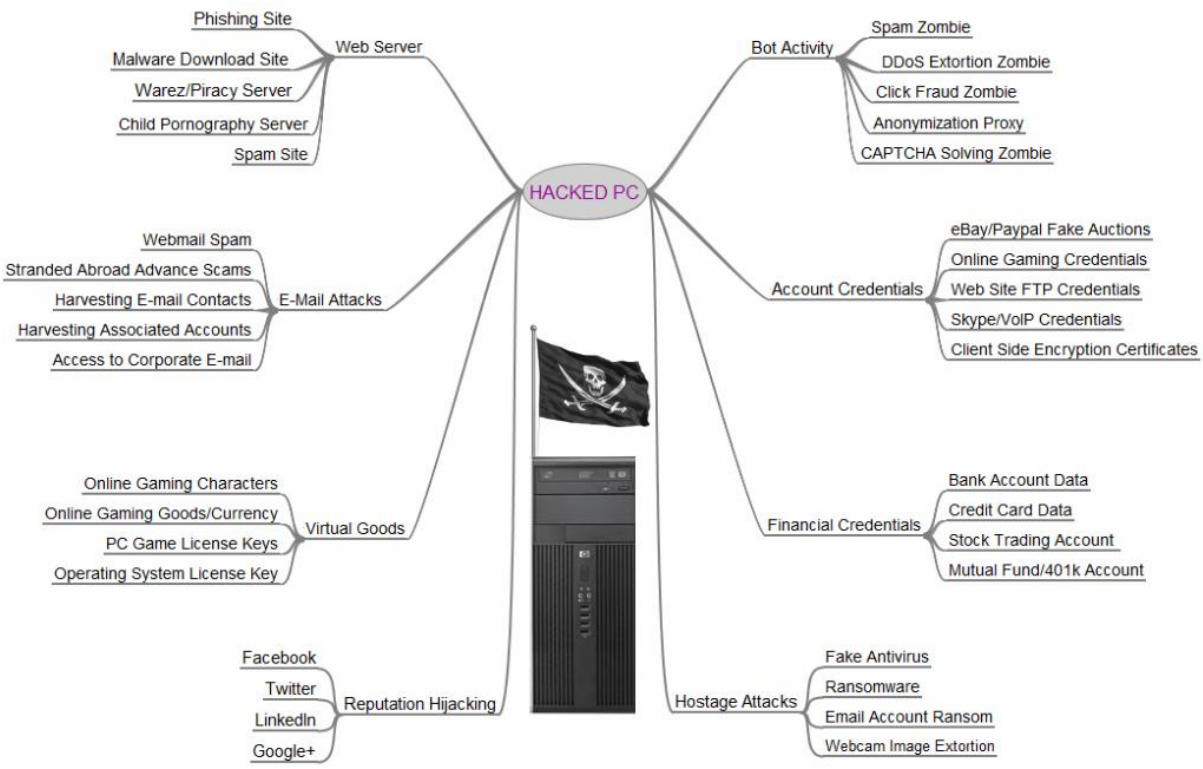


Figure A: Why someone would want to hack your PC, by Krebs
http://krebsonsecurity.com/2012/10/the-scrap-value-of-a-hacked-pc-revisited

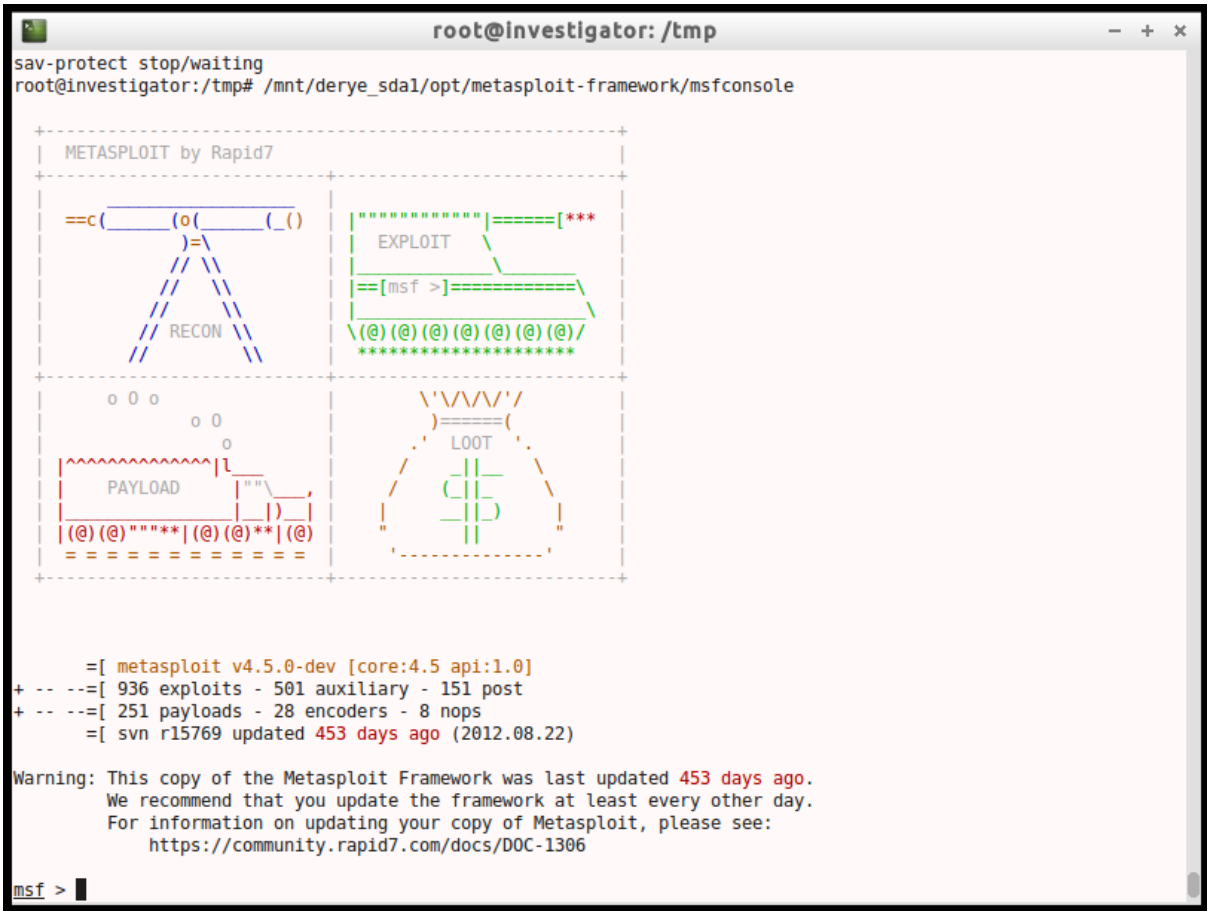


Figure B: A Metasploit attack process. Screenshot of Metasploit by Rapid7

Recovering evidence from the suspect's machine

Past cases show all of the following are potentially retrievable from a suspect partition (provided that they exist).

Evidence	Retrievable	Timestamped
Browser (e.g. Mozilla Firefox):		
URL History	yes	yes
bookmarks	yes	yes
downloaded files	yes	yes
cached webpages & images	yes	yes
saved passwords	yes	-
history of password submissions	yes	yes
saved cookies	yes	yes
e-certificates	yes	yes
installed add-ons	yes	yes
Email client (e.g. Mozilla Thunderbird):		
received messages	yes	yes
sent messages	yes	yes
deleted messages	possibly	yes
email headers / source IPs	yes	yes
saved passwords	yes	yes
clicked URLs (including attachments)	yes	yes
installed add-ons	yes	yes
terminal command history (e.g. Bash)	yes	not by default
Metasploit command history	yes	no
downloaded utilities	yes	yes
SSH saved server keys	yes	-
keyring manager contents	yes	-
instant messenger accounts	yes	-
database logs (e.g. postgres)	yes	yes
DHCP connection logs	yes	yes
Window manager GUI		
history of opened files	possibly	yes
history of GUI editor recent files	usually	yes
preferred wallpaper	yes	-

cached image thumbnails	yes	-
operating system:		
distribution version	yes	-
hostname	yes	-
timezone	yes	-
order of all file modification	yes	yes
order of all file access	yes	yes
extra partitions table (`fstab`)	yes	-
operating system & hardware logs (`dmesg,messages,user,lpr,mail`) e.g. external USB drives plugged in, past printer queue, OS alerts	yes	usually
remote network filesystem mounts	yes	yes
user accounts and groups	yes	-
order of addition	yes	-
user account passwords	yes	-
deleted accounts (backup file)	possibly	-
cached printer files	yes	yes
cached OS mail	yes	yes
cached IP addresses quickly e.g. `egrep -r "[0-255]{1,3}\.[0-255]{1,3}\.[0-255]{1,3}\.[0-255]{1,3}" /var/`	yes	-
played sounds log	yes	yes
remote database connections	yes	yes
Garmin geolocation log	yes	yes
Local user directories	yes	yes
stored public/private keys (G/PGP)	yes	yes
file access times	yes	yes
file modification times	yes	yes
deleted file listings	yes	yes
deleted file content	possibly	yes
Bootable Live Session of image	possibly	-

Much of this information, such as searching the filesystem for IP addresses, server keys, and website/server credentials can potentially lead to the seizure of online accounts - and allow us to trace the suspect's parent organization and contacts.

We can retrieve data from any number of applications (even if the data is password protected usually the only factor is time – it depends on the strength of the encryption).
Depending on the operating system there is often an individual application profile for each user account (stored in their home directory).

In the case of operating system information (such as a system's users), we sometimes find these grouped for all accounts (such as in the `/etc` folder).

If we are quick enough to `dd` the image of a live system, we can capture volatile system information and shared cache files (such as in `/var`, `/tmp` or `/proc` directories). These can be particularly useful in a live system because it can contain present state information, like the contents of RAM and current user or network sessions. It is vital that one images these as soon as possible, before the state changes and becomes redundant.

Part 2: Investigation report

Contents

Total: 38 pages

MPANT electronic-only.pdf *contains:*

Assumptions:
Shell history was not retroactively modified (no 'covering footprints' phase);
The laptop is not part of a virtual instance;
Deleted emails (if any) are non-retrievable, and cannot sway the validity of the stated evidence;
Appropriate warrants from third parties have been approved;
No valuable user account settings (or associated home directories) were removed.

For readability scripts and comments have been colour syntax highlighted.
For convenience Listing A has been mirrored in MPANT electronic-only: Appendix C so that you may keep a colour copy on screen to prevent having to flip back and forth.
A lot of time has gone into relevant inter-referencing, so gaining a familiarity with **Appendices A-C** prior to reading should make the review more fluid and enjoyable.

Introduction

I have been retained as a forensic computer analyst by the prosecutor of the King Charles Government. My purpose is to analyze a laptop image seized from *Alexandrine de Rye* (Countess of Thurn and Taxis) by Officials at the Belgium/Netherlands border in August 2012. With the evidence garnered my intention is to assert whether the suspect may or may not have been guilty of cybercrime, specifically that pertaining to an incident recognized on August 25th regarding the defacement of *pepys.dyndns.org* (owned by one *Mr. Samuel Pepys* of the Kings Charles Government, Royal Navy office, London).

Background

Owing to tensions caused by the Anglo-Dutch war, our Government has become a target for cyber-terrorists from opposing nations and their allies.

We have possession of a laptop confiscated from *Alexandrine De Rye*, a security specialist who is a servant of *Elizabeth Queen of Bohemia*.

Since we know her ruler to be a supporter of the Dutch, we have reason to suspect the captured device may contain evidence relating to the case.

Sequence of Events

1) On August the 22nd 2012 at 02:04:41 CEST "*Elizabeth Queen of Bohemia*" (holder of *elizabeth@oxfordian.info*), part of the *oxfordian.info* organization of which *Alexandrine De Rye* is also a member, emailed the suspect. She was given information about the victim, and instructed to collect further information from the now vandalized server (*pepys.dyndns.org*) [Appendix A: Article-A1].

2) A day later on August the 23rd at around 01:26:02 CEST the suspect began to use her laptop [Listing A: Alex's timeline: (line number) 1].

3) The suspect queried her email server (*imap.googlemail.com*) at 02:05:21 CEST, checking the account of "*Alexandrine de Rye*" (holder of *elizabeth@oxfordian.info*) at which point she would have found this message. [Listing A: Alex's timeline:99] indicates this, provided she had not checked her online account previously).

4) Following this at 02:07:05 CEST the suspect loaded her browser [Appendix B: Browser history:4]. At 02:07:22 CEST evidence shows she was browsing a guide on how to install Rapid7's "***Metasploit" security pentration and exploitation tool***" on an operating system exactly matching her own. When we review the earliest modification time of the `local opt/metasploit-framework` directory [Notes: Filesystem listings: export ID 1] we believe the tool was downloaded entirely at once at 02:59:04 CEST the same day.

5) We believe that her operating system was upgraded, and a number of additional applications were installed within the hour: `subversion`, `vncviewer`, `postgresql` and a number of dependent libraries [Listing C: bash history].

6) At about this period 02:49:53 CEST [Listing A: Alex's timeline:118] she entered a local `/home/alex/Development` directory and downloaded the `nmap` security penetration testing tool.

7) We believe that she performed non-digital reconnaissance between the 23rd and 25th of August 2012 [Listing A: Alex's timeline:4736-4737].

8) On the 25th at about 02:47:27 CEST she returned to her laptop [Listing A: Alex's timeline:4737].
Interspersing her webrowsing with the use of her email client, two documents of username and password lists were modified (`home/alex/hacks/user-list.txt` at 02:54:30 CEST, & `home/alex/hacks/pass-list.txt` at 02:55:00 CEST), which were utilized by the Metasploit program [Listing B: Metasploit history].

9) She took some interest in a VPN server provider at 03:04:34 CEST [Appendix B: Browser history:5]; an additional IP address she had saved in her SSH hosts (and connected to with `posgresql` as her operating system `var/ cache`

indicated) was 96.234.173.244 – matching Elizabeth Queen of Bohemia's email header IP address, it also may be an SSH proxy server used by the suspect (to launch the attack).

10) At precisely 18:05:43 CEST on this day she browsed to the victim's site for the first time [Appendix B: Browser history:9]. It is during the next 4 hours that we see sustained browser and filesystem activity, and we believe the breaches took place. From reading the cache, we are able to state the content of the sites that were visited.

11) She took particular interest in the page http://pepys.dyndns.org/?page_id=2 [Appendix B: Browser history:10], visiting it 9 times during the course of the attack.

12) We breakdown her browser activity in detail when we analyze the cache in [Listing A: Alex's timeline].

We will summarize it here. The suspect appears to research numerous Metasploit utilities between 25/08/2012 18:12:25-19:04:03 CEST, trying them one by one on the victim's site [Appendix B: Browser history:12-19] — having already downloaded the utilities at once days earlier (indicated by sequential chronological timestamps).
We note that during this period (at 18:59:58 CEST) the SSH known hosts file was modified [Listing A: Alex's timeline:4872]. Since the latest entry is the second server private key (verified as 96.234.173.244), it suggests SSH was used to connect to this server at this time. Note this SSH host matches the originating IP address of [Appendix A: Article-A1] from Elizabeth Queen of Bohemia.

13) The suspect logged on to the victim's administration control panel successfully at 19:11:58 CEST.
The suspect browsed posts, profiles and themes (in line with her reconnaissance mission) [Appendix B: Browser history:29-36]. She was logging in and out repeatedly, perhaps to test different logins and use escalated access permissions.
Our cache later indicates the presence of an 'alex' (belonging to 'alexking.org') in the Wordpress control panel.

14) At 19:18:44 CEST she began to research the Metasploit MySQL utility [Appendix B: Browser history:29-37].
We believe the utility (as with the others) was launched against pepys.dyndns.org (supported by [Listing B: Metasploit history].

15) The suspect returned to her email client at 20:49:37 CEST [Listing A: Alex's timeline:4931], emailing [Appendix A: Article-A2] at 21:02:56 CEST to the Queen.

16) "Elizabeth Queen of Bohemia" returned [Appendix A: Article-A3] to the suspect, stating her pleasure, and a suggestion that the blog be altered to show support for the Netherlands. An image of the Netherlands flag (`200px-Prinsenvlag.svg.png`) was included, which was later found on the victim's server.

17) Following checks of her email (after browsing Belgian news site http://www.nieuwsblad.be), the suspect recommenced browsing http://pepys.dyndns.org at 21:19:06 CEST [Appendix B: Browser history:47].

18) At 21:19:14 CEST the suspect logged into the victim's administration control panel [Appendix B: Browser history:49].

19) During the session of 21:19:14 CEST to when the suspect logged out at 21:33:11 CEST is when we believe the suspect uploaded the offending image.
Our browsing history illustrates the time at which the attachment was uploaded (21:29:23 CEST, [Appendix B: Browser history:70]) via the blog control panel and verified online by the suspect (21:31:31 CEST, [Appendix B: Browser history:76]). Supporting the browser logs we see various cached images on the victim's harddrive of the pages that were visited. Most notably, [Listing A: Alex's timeline:4978] cached at 21:32:26 CEST shows an image of the panel being edited with the words "*support Republiek der Zeven Verenigde Nederlanden!*".

20) We see a browser cached file at 21.33:55 CEST (19:33:55 UTC) suggesting that the page was modified at 2012-8-25 8.33pm [Listing A: Alex's timeline:4981]. We expect to confirm with Mr. Samuel Pepys that the site was set to BST (UTC+1) during this period.

Alex then appears to terminate the browser and email client. Here ends the sequence of events.

Evidence

Environment reconnaissance

Operating system *Notes: Environment reconnaissance: operating system*

The image is of a `Linux` operating system partition.

The bootloader configuration file (`boot/grub/grub.cfg`) suggests the distribution and kernel version is "`Ubuntu, with Linux 3.2.0-29-generic-pa`".

A Linux Standard Base configuration file (`etc/lsb-release`) identifies the distribution as "`Ubuntu 12.04.1 LTS`". This is confirmed by a system identification configuration file (`/etc/issue`).

It is based on the Debian Linux distribution release '`wheezy/sid`' (`etc/debian_version`), '`sid`' referring to an "unstable" development branch.

Hostname *Notes: Environment reconnaissance: hostname*

Standard hosts files (`etc/hostname, /etc/hosts`) display the system hostname as 'hacker-box'.

Timezone *Notes: Environment reconnaissance: timezone*

The system has two timezone files (`/etc/timezone` (plain-text), `/etc/localtime` (encrypted but readable)).

The first of the files states the timezone is set to 'Europe/Brussels'; importing the latter appears to verify this.

Clock calibration *Notes: Environment reconnaissance: clock calibration*

We have a choice - we can either configure our clock to present our results in UTC;
or we can present our results in CEST (the timezone of "Europe/Brussels" which was valid March 25 2012-October 28 2012 : UTC+2 hours).
Having performed both exports, we have chosen that presenting the results in CEST (the timezone the evidence correlates to, as well as the circumstances of the suspect's capture) is easier to interpret when reading this report.

Therefore, our host system has the timezone set to CEST and synchronized with NTP servers online;
The date & time were repeatedly verified as being accurate to the 'current' CEST time (within microseconds)[1]

This results in all of our inode timestamp readings being absolutely consistent to each other (to the second).

Users *Notes: Environment reconnaissance: users*

Before we assert the suspect's chosen user account(s) for the attack, we must confirm all user accounts (present or deleted) of the system.
The standard 'accounts' file (`/etc/passwd`) lists these all.

We found two user accounts to investigate: 'root' (`/root`) and 'alex', named 'Alexandrine de Rye' (`/home/alex`).

We find no suspicious account deletions when comparing with a backup file; we have assumed that no valuable accounts were deleted.

[1] Network Time Protocol project, http://www.ntp.org/ntpfaq/NTP-s-sw-clocks-quality.htm

Applications of interest

Email client *Notes: Applications of interest: email client*

Email correspondence (under an account named "Alexandrine de Rye" both at the local client and at the online Google Inc. account, of the email address alexandrine@oxfordian.info) has been retrieved.

The email correspondence shows that the 'account owner' had engaged in written conversation with an email account named "Elizabeth Queen of Bohemia" (elizabeth@oxfordian.info), an address registered to the same internet domain (and therefore likely to have a common domain administrator and parent institution).

The original itemized thread of this conversation can be found in [Appendix A].

Message content

Article-A1: On the 22/8/2012 (UK date format) at 00:04:41 UTC (02:04:41 CEST), "Elizabeth Queen of Bohemia" initiated the conversation with "Alexandrine de Rye" (message subject entitled: "Samuel Pepys & The British Navy").

Her email states:
"Elizabeth Queen of Bohemia" was "*in support of the Dutch against the British*"
"*we*" (either the Queen or Bohemia) wanted to "*find out more about Samuel Pepys' activities at the British Navy Office*", suggesting an information reconnaissance mission.
She was aware that "Mr. Pepys" had "*just set up his own server for email and blog services*" at "*Pepys.dyndns.org*".
She was instructing "Alexandrine de Rye" to "*find out who Mr. Pepys is emailing and what their correspondence is about*"
An information reconnaissance mission was already underway; through the practise of 'social engineering' members of the Queen's chamber had a list of names of "*Mr. Pepys' acquaintances*".
"Samuel Pepys" (who we know to be the owner of Pepys.dyndns.org) and 7 acquaintances are listed - these acquaintances should be confirmed by our victim.

Article-A2: On the 25/8/2012 (UK date format) at 21:02:56 CEST "Alexandrine de Rye" responded (message subject entitled: "Re: Samuel Pepys & The British Navy").

Her email states:
She had "*performed*" some activity "*Mr. Pepys' website, pepys.dyndns.org*"
She verifies "Elizabeth Queen of Bohemia"'s reconnaissance information as being "*correct*"
She establishes use of "*pepys.dyndns.org*" to "*exchange emails*" between "*Samuel Pepys*" ("*samuel@pepys.dyndns.org*") *with "Mr. John Evelyn*" ("*john@pepys.dyndns.org*").
She includes an attachment '`mail-thread.txt`' which contains email communication between our victim and "*Mr. John Evelyn*" ("*john@pepys.dyndns.org*") - this should be confirmed by our victim.

She analyses and summarizes the content of this attachment, relaying information about the battle environment, the position and status of combatants, British expenditure and opinions held by our victim ("*Mr. Pepys is concerned about the fate of British sailors shipwrecked in Ireland following a battle with the Dutch..the British are in no rush to rescue their sailors..Mr. Pepys is concerned for the fate of prisoners that the British have taken..the prisoners are ill and not being treated..concerned promised funds are not being provided to hold the prisoners as the money is being diverted* ").

She admits belonging to the "*Black Chamber*" group. The email implies she and her division are the Queen's "*obedient servant*".

Article-A3: On the 25/8/2012 (UK date format) at 21:16:27 CEST "Elizabeth Queen of Bohemia" responded (message subject entitled:"Re: Samuel Pepys & The British Navy").

Her email states:
pleasure in "Alexandrine de Rye"'s previous transmission;

an implication that for "Alexandrine" to perform successful operations is "*usual*" (this might suggest that "Alexandrine" may be a computer security breach professional and have prior offences that have been commanded by the Queen).
The information transmitted is relevant and shall influence Dutch strategy positively ("*Your news comes at a significant time for our Dutch allies. With such disarray rescuing sailors and a shortage of funds it seems one final push may lead to victory.*")
An implication to 'mudrake' "*Mr. Pepys integrity*" by altering "*his blog to show his support for the (Republic of the Seven United Netherlands)! Our allies flag is attached*".

She includes an attachment '`200px-Prinsenvlag.svg.png`', a graphical image of the original "Prinsevlag" flag, which has tested negative for concealed steganographic information.

Headers

The header data of these messages reveals further IP address information of the outgoing and incoming mailservers, as well as the originating and destination computers. We can utilize this to verify and potentially seize these machines, so as to support our assumptions of the validity of our case evidence.

All messages from "Elizabeth Queen of Bohemia" share the following line of interest in the header:
`X-Originating-IP: 96.234.173.244` which illustrates the client's device IP address.
Entering this IP address into a geolocation tracer[2] identifies the sender's originating city as Columbia, Maryland, U.S.A., and their ISP as Verizon FiOS. These leads us to 3 possible assertions:

- That "Elizabeth Queen of Bohemia" and her device were based here;
- That only the sender's client device was based here (e.g. she may have remotely accessed this from another country, especially if using a hosted solution such as a remote VPN server);
- That I.P. address spoofing software was used to falsify this record.

Subsequent header values suggest that the email hosting provider was Google Incorporated.

The message from "Alexandrine de Rye" has the following lines of interest in the header:
```
Received: from [172.16.77.129] ([213.179.209.92]) by mx.google.com
User-agent: Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20120714 Thunderbird/14.0
```

With the same cautious assumptions as before, we enter these IP addresses into a geolocation tracer which identifies the sender's originating city as Amsterdam, Netherlands (where `172.16.77.129` may be an internal private Class B network address, and `213.179.209.92` the external) and their ISP as SolidHost.

We also note the operating system and local mail client application configuration displayed, which matches the environment configuration of the seized laptop.

Online account
As well as header information that indicated routing to Google Inc. servers, the first email present in the client's mailbox was received at 08/22/2012 01:38:27 CEST (08/21/2012 23:38:27 UTC) from "Gmail Team","mail-noreply@google.com" (message subject entitled: "Get started with Gmail"). This is a standard email sent from "Google Apps" (*http://www.google.com/enterprise/apps/business/)* when a user account is created on an existing domain. We have no suspicion that this was forged, and hence Google Inc. hold an account for the domain "oxfordian.info". Both "alexandrine@oxfordian.info" and "elizabeth@oxfordian.info" belong to this account.

Loading a working copy of 'alex''s profile into the same application release as used by the client (Mozilla Thunderbird 14.0, http://mozilla.org) we are able to verify the profile settings.

We note that the account has been configured to use IMAP to connect to the parent Google Gmail account.
IMAP means that all client applications and devices configured to use IMAP are all synchronized to one folder tree.
E.g. if the client were to delete an email in their local application (here "Thunderbird") the email would also be

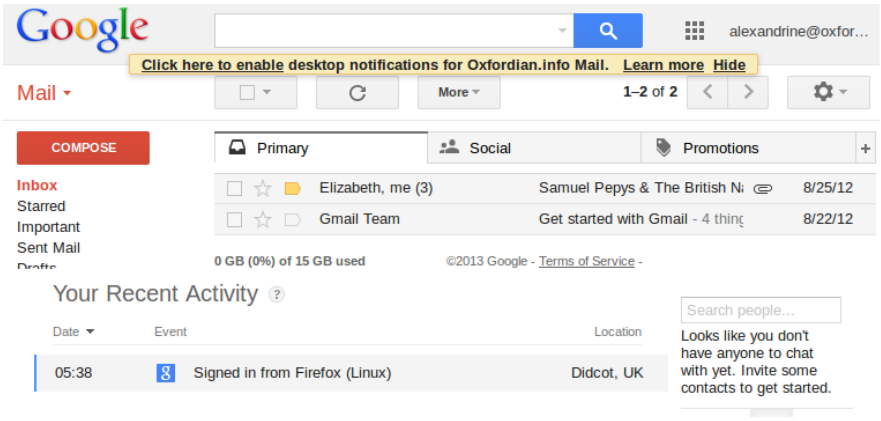
[2] in this example we used http://www.iplocationfinder.com for lack of verified tools

deleted from the Gmail Trash folder at the next synchronization (in 'alex''s profile 'h9jaeoos.default' this is set to every 10 minutes).
Were POP used instead of IMAP the local client's folder tree would not be automatically synchronized with the parent Gmail account, and we would want to investigate the hosted account to view this discrepancy (for example, even if a user had deleted a message from their "Thunderbird" Trash box and their Gmail inbox, had they been using POP their email would still be present in a Gmail Trash folder for 30 days).
Even if using IMAP, the Google account may harbour data that the Thunderbird account does not.

As thorough investigators we still suspect that the Gmail account may contain additional information.
Via Mozilla Thunderbird 14.0 we are able to retrieve the saved account credentials from this profile easily.

username	password
alexandrine@oxfordian.info	rynLopHoj9

Since Google comply with Government agency requests for account access (*http://www.google.com/transparencyreport/userdatarequests/legalprocess/#why_might_a_government*), we have assumed they have created a working copy of this account for us to interrogate. In a real world situation the next phase of investigation could not be achieved without a warrant, as trespassing on a third-party site is an infringement of law, and an act of altering the evidence of server logs.



Checking for additional login locations and contacts

Having gained access we want to verify several things:
Whether the online mailbox contains any messages or labelled items that may not be present in the "Thunderbird" profile (e.g. archived items);
If we can pinpoint other contacts (sharing the domain or otherwise) that can lead to more evidence;
If there is a log of their chat history (if any);
If we can view the IP addresses and locations from which the suspect logged into their online account;
If the user has set up any mail forwarding or mailbox filters (i.e. processing rules, such as 'skip inbox and forward') that would intercept messages from being downloaded to Thunderbird (and lead us to additional suspect accounts);
If the user has made use of any of the other facilities that their account offers (e.g. for storing shared documents, maps, their calendar, their Google+ profile, their groups, etc. - anything that could lead us to more evidence).

Having searched this online account fully we see no additional evidence, except that:

The password was "Changed over 1 year ago on 21 August 2012", either by Alex or the site administrator - this may also signify the creation of the account (it matches the date of the welcome message from the "Gmail Team");
The recent activity log shows only *our* login (IP address, date and location), so the user does not appear to have logged in online recently (and we have a suspicion that no one has logged in since at least June – *discussed in* [Notes: Applications of interest: Email client]);
That the inbox welcome message may lead us to the network administrator[3]; we might retrieve more of the suspect's network or VPN internet activity by retaining them.

[3] *"created by Gareth Digby"*

Internet browsing history of a local client application has been retrieved. One profile was found to be active in the suspect's eponymous user directory.

The full itemized history can be found in Appendix B [Appendix B: Browser history].

The URL log

Computer account 'alex''s browser history begins on 23/08/2012 (UK date format) 02:07:13 CEST local time.
It ends upon 25/08/2012 21:34:30 CEST.

This browsing activity begins after the initial message [Appendix A: Article-A1] from "Elizabeth Queen of Bohemia", and ends 1 hour and 18 minutes after message 2/2 [Appendix A: Article-A3] from "Elizabeth Queen of Bohemia". Provided that 'alex' had read these messages, we may infer that subsequent browsing was performed with those requests in mind. Let us attempt to correlate "Elizabeth Queen of Bohemia"'s requests with the sites visited.

[Appendix A: Article-A1]: On the 23/8/2012 (UK date format) at 02:04:41 CEST, "Elizabeth Queen of Bohemia":
Introduces "*Mr Pepys*"and "*Pepys.dyndns.org*".
Requests information gathering to be performed on "*blog*" and "*email services*".

[Appendix B: Browser history]: 23/08/2012 02:07:14 CEST, browser of account "Alexandrine de Rye":
Line 1: Views an article about "*Installing Metasploit Framework on Ubuntu 12.04*" (suspect's operating system).
The `'metasploit-framework'` is identical to that found in the seized partition's `/opt/metasploit-framework/` directory. This implies research with intent.

The operating system mentioned is 'Ubuntu 12.04'. While this matches the suspect's operating system, the suspect's reconnaissance work (`/home/alex/hacks/pepys.dyndns.org.reconn.txt{~}`) does not lead us to believe this was the operating system of the victim's server. We must confirm this statement with Mr. Samuel Pepys, and deduce whether the research conducted by the suspect would have been applicable to pepys.dyndns.org, and whether the server was probed by this same version of `'metasploit-framework'` found on the suspect's machine.

At this point the user takes a hiatus from browsing; both section [Notes: Filesystem listings: export ID 1] ***&*** [Listing B: Metasploit history] ***suggest she downloaded and installed Metasploit into /opt/metasploit-framework/***

25/08/2012 03:04:34 CEST:
Line 5: Visits a site for creating a private encrypted tunnel to a proxy server. The user does not appear to have logged into any account, but viewed the landing page and reviews. Since a VPN client browses via a computer in a remote location (this company offers servers in 19 countries, including Amsterdam, Netherlands), any log records found on Pepys.dyndns.org traced to this company may give clue of an account held by the oxfordian.info organization. Such an account would be retrievable with a warrant as the site complies to international law (http://strongvpn.cc/policy.shtml).
That the user specifically visited the .nl domain (when strongvpn.com holds multiple TLDs, including .cc, .com) might imply the user's country preference, or a proximity redirection based on device network location.

25/08/2012 17:48:26 CEST:
Lines 7-8: Visits a Belgian news website (http://www.nieuwsblad.be). Views an unrelated article written in Dutch. The sequence of the previous 3 websites might suggest that a VPN proxy has become active (browsing from the Netherlands to now Belgium), or may just be the client's URL preference.

25/08/2012 18:05:43 CEST:
Lines 9-19: The suspect intersperses visits to Pepys.dyndns.org (and repeatedly "page_id=2") with 3 pages showing Metasploit documentation. [Listing B: Metasploit history] ***suggests all modules were downloaded at once via SVN on August 23rd, so these visits were like for researching documentation.***

websites: http://pepys.dyndns.org/*
25/08/2012 18:12:25 CEST: http://www.metasploit.com/modules/auxiliary/scanner/portscan/tcp

25/08/2012 18:13:39 CEST: http://www.metasploit.com/modules/auxiliary/scanner/ssh/ssh_login
25/08/2012 19:04:03 CEST: http://www.metasploit.com/modules/auxiliary/scanner/http/wordpress_login_enum
25/08/2012 19:11:58 CEST: http://pepys.dyndns.org/wp-login.php

We must check the webserver logs of pepys.dyndns.org to see if these tools probed the server at a similar time to the suspect's research.
Viewing the "*Wordpress Authentication Brute Force and User Enumeration Utility*" is followed by a visit to what appears to the victim's *Wordpress* login page. We must confirm this statement with Mr. Samuel Pepys.

25/08/2012 19:11:58 CEST:
Lines 20-36: The chronological order and stubs of the webpages visited during these 4 minutes suggest "Alexandrine de Rye" successfully logged in to the admin panel, visited administration pages that can be used for editing a post, editing page links, and editing a profile, and logged out. We must confirm this statement with Mr. Samuel Pepys, and correlate the modification times of the altered files on the webserver with these visits (provided that Mr. Samuel Pepys has not altered the evidence), and confirm which username(s) was/were used.

25/08/2012 19:18:44 CEST:
Lines 37-43: Suspect visits a module tutorial page for the "*MySQL Login Utility*" for bruteforce interrogating a MySQL database of its username and password.
This is followed by viewing the documentation for MySQL that runs on (or is compatible with) the 'Ubuntu 8.04 Hardy' operating system, followed by revisiting http://pepys.dyndns.org/?page_id=2.
We must confirm with Mr. Samuel Pepys whether this version of MySQL documentation matches or was compatible with that which ran on his server (the answer to this question will affect the weight of this evidence), and whether "page_id=2" of his blog was breached with any of the aforementioned modules.

25/08/2012 20:49:14-21:14:05 CEST
Lines 44-46: Alex visits http://www.nieuwsblad.be thrice within 25 minutes, during which she sends [Appendix A: Article-A2].

[Appendix A: Article-A2]: On the 25/8/2012 (UK date format) at 21:02:56 CEST "Alexandrine de Rye":
Admits she had "*performed some wizardry [actions] on Mr. Pepys' website, pepys.dyndns.org*".
This supports the view that the 'alex' user account's browser profile is held by the same "Alexandrine de Rye" of alexandrine@oxfordian.info. We would like testimony from "Alexandrine de Rye" that this statement is correct.

Reveals details about the use of "*pepys.dyndns.org*" to "*exchange emails*" between "*Samuel Pepys*" of samuel@pepys.dyndns.org with "*Mr. John Evelyn*" of john@pepys.dyndns.org.
She includes an attachment `'mail-thread.txt'` that we believe to have been taken from the server. We must confirm this statement with Mr. Samuel Pepys, and whether his logs show that the intrusion times coincide with 'alex@hacker-box''s times of activity.
We would also like to confirm with Mr. Samuel Pepys if the originating IP addresses of the attack match any of the IP addresses previously found in 'Alexandrine de Rye''s email headers, particularly 213.179.209.92/172.16.77.129, or Elizabeth's SSH server, or any belonging to a VPN account owned by StrongVPN.nl.

We would like to acquire testimony from our suspect and/or her email domain administrator that the email address "alexandrine@oxfordian.info" was assigned to her, and that she is the owner of the 'alex' laptop user account (as the full name field implies), and that she was sole owner of the browser profile `'wbaoqjsm.default'`.

[Appendix A: Article-A3]: On the 25/8/2012 (UK date format) at 21:16:27 CEST "Elizabeth Queen of Bohemia":
Suggests altering "*his* (Pepys') *blog to show his support for the* (Republic of the Seven United Netherlands)!";
Includes the attachment `'200px-Prinsenvlag.svg.png'`.

[Appendix B: Browser history]: 25/08/2012 21:19:06 CEST, user account "Alexandrine de Rye":
Line 47: Following the 25 minute interlude and new instructions, 'alex' returns to http://pepys.dyndns.org.
Between lines 47-91 'alex' appears to conclude a final second attack on the target.
Whereas the first mission was for retrieval, the second is for modification.

The chronological order and stubs of the webpages visited during this time suggest 'alex' successfully logged in to the admin panel, visited administration pages that can be used for editing a new page, editing a post, editing a theme, editing page links, editing a profile, uploading an image, checking uploads, and posting an upload.

In line 25: 25/08/2012 19:12:34 CEST: we saw one reference to wp-admin/edit.php; a page that could potentially be used to edit. Now, between lines 47-91, we see the wp-admin/edit.php followed by wp-admin/edit-pages.php, post.php?action=edit&post=2, page-new.php, link-manager.php, etc. It seems that she is proceeding with the modifications. Let's interpret and summarize this "second wave" of URLs listed chronologically (which we would like to verify with Mr. Samuel Pepys):

Line 49: 25/08/2012 21:19:14 CEST: There are three action=logout lines in the entire log; therefore this point is the third and final time she logs into the administration panel (having logged in and out twice previously).

Line 52: 25/08/2012 21:19:42 CEST: she visits page_id=2, and

Line 55: 25/08/2012 21:19:59 CEST: stub wp-admin/edit-pages.php is seen for the first time.

Line 56: 25/08/2012 21:20:06 CEST: post.php?action=edit&post=2: she appears to edit post=2

Line 57: 25/08/2012 21:21:26 CEST: interest in a source image for a theme (themes/advanced/image.htm?src=)

Line 59: 25/08/2012 21:22:15 CEST: may be attempting to create a new page (wp-admin/page-new.php)

Line 62: 25/08/2012 21:22:21 CEST: visits the control panel for links (wp-admin/link-manager.php)

Line 67: 25/08/2012 21:22:54 CEST : post.php?action=edit&post=2: we see the second of 3 of these URLs (submitting an edit to post=2) which may represent "page_id=2" – confirmation requested from Mr. Samuel Pepys

Lines 70-72 25/08/2012 21:29:23 CEST: an upload appears to take place (wp-admin/inline-uploading.php?action=upload&post=2&all=&start=0), followed by an apparent confirmation that a modification was posted (wp-admin/post.php?posted=true)

Lines 75-76: 25/08/2012 21:31:31 CEST: An uploads page is visited (http://pepys.dyndns.org/uploads - not necessarily administration), and the filename that 'alex' received in [Appendix A: Article-A3] is now online at http://pepys.dyndns.org/wp-content/uploads/2012/08/200px-Prinsenvlag.svg.png merely 15 minutes and 4 seconds after it was sent to the suspect.

Lines 78-79: suspect visits http://pepys.dyndns.org/?p=10 and http://pepys.dyndns.org/?p=8

Line 81-87: 25/08/2012 21:32:23-21:33:03 CEST: A final set of editing appears to take place. Cached files found in the suspects browser directory that follow will shed light (with visual images which are timestamped) of precisely what pages the suspect was viewing and modifying throughout their visits to pepys.dyndns.org

Line 81-87: 25/08/2012 21:33:11 CEST: The suspect finally logs out just 13 minutes and 57 seconds after logging in. We hope to confirm that the attack took place between these reviewed hours by testimony from Mr. Samuel Pepys

Line 90-91: 25/08/2012 21:33:14 & 21:34:30 CEST: the suspect views pepys.dyndns.org and http://pepys.dyndns.org/?page_id=2 from the front-end once more.

This concludes our account of the browser history log.

Bookmarks, Downloads, and other databases *Notes: Applications of interest: internet browser*

Analyzing the bookmarks of this profile reveals nothing unusual. There is a reference to '*Ubuntu One*' - an online storage Cloud, but this appears to be part of the standard distribution for the suspect's operating system. More supporting trace evidence is required before any online account is investigated.

Any files that may have been downloaded have been cleared from the profile's logs.

Form submissions:
Logs indicate that the client application twice submitted a form at http://pepys.dyndns.org:

URL of submitted form	user	password	time created (not literal)	time last used (interval)	Times password changed	times used
http://pepys.dyndns.org	*log*	*pwd*	*1970-01-16 09:51:54*	*1970-01-16 09:52:02*	*1970-01-16 09:52:02*	*2*

We additionally retrieve a saved login from the client profile:

Site	Username	Password
http://pepys.dyndns.org	*Samuel*	*sarah*

And this matches some of the credentials stored in the 'alex' home folder (discussed subsequently).

We must confirm with Mr. Samuel Pepys whether these were breached administrative logins used to modify the pages affected at the times above, and whether either of these credentials were also changed.

Instant Messenger *Notes: Applications of interest: instant messenger*

Comparing the public keys found in a folder commonly used by an Instant Messenger application (`usr/lib/purple-2`) against our baseline system, nothing differs from standard operating system distribution files.

Secure Shell (SSH) *Notes: Applications of interest: SSH*

We find an SSH configuration file (`.ssh/known_hosts`) verifying 'alex' had saved trusted public keys from pepys.dyndns.org and 96.234.173.244, and thus connected previously to both via the SSH protocol.
The latter address matches Elizabeth Queen of Bohemia's email header, and we seek identification of the latter IP address in Mr. Samuel Pepys's affidavit to confirm whether it was used as a proxy in the attack.

GNOME Keyring
(credential, certificate and key manager) *Notes: Applications of interest: keyrings*

We find a "GNOME Keyring" database file (`.gnome2/keyrings/login.keyring`) which contains a saved password ("black") which we extracted. Unfortunately this did not unlock further relevant account information. We additionally unlocked the `/etc/passwd` file, stating the 'alex' username as having password "black" allowing us to proceed with any interactive "Live View" analysis if required.

gedit (GUI text editor) *notes: Zeitgeist and gedit (GTK+)*

We find a GTK+ application history file (`.local/share/recently-used.xbel`) which is used by "gedit".
Since the log file is also used by the Zeitgeist activity logger, we present and correlate these files and timestamps pictorially together in the following section GUI history: Zeitgeist and gedit (GTK+).

Filesystem directories of interest

Exploring user home directories

The 'root' account home directory (`root`) *notes: Filesystem listings*

We found a trace of cache access (a sound file was played; likely an echo of a previous login), but nothing significant to enter into court.

The 'alex' account home directory (`home/alex`) *notes: Filesystem listings*

A **colour coded**, **trimmed** and **commented** timeline was derived from [Notes: Filesystem listings: export ID 5].

Every line has been scrutinized for clues, and appropriate descriptions have been added.
This incorporates discussion of the suspicious files the user created.

Listing A: Alex's timeline

Analysed timeline of 'alex' (UID 1000): `home/alex`

Screen copy: [Appendix C: Listing A]

Unmodified version available online (139 pages)

See Notes: Filesystem listings

Activity Guide
affected by environment layer applications (e.g. ubunutu OS, window manager, terminal
affected by user 'Mozilla Thunderbird' activity
affected by user 'Mozilla Firefox' activity
affected by additional application activity
directly created or modified by 'alex'
Email message sent
Comment

↘'alex' begins the account session

1 2012-08-23 01:26:02 home/alex/.bash_logout
2 2012-08-23 01:26:02 home/alex/.bashrc
3 2012-08-23 01:26:02 home/alex/.profile
4 2012-08-23 01:26:02 home/alex/examples.desktop

↘utilizes the password manager

5 2012-08-23 01:32:12 home/alex/.gnome2/keyrings/login.keyring
6 2012-08-23 01:32:12 home/alex/.gnome2/keyrings/user.keystore
7 2012-08-23 01:32:12 home/alex/.config/user-dirs.dirs
8 2012-08-23 01:32:12 home/alex/.fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-le32d4.cache-3
9 2012-08-23 01:32:12 home/alex/.config/user-dirs.locale
10 2012-08-23 01:32:12 home/alex/.fontconfig/e13b20fdb08344e0e664864cc2ede53d-le32d4.cache-3
11 2012-08-23 01:32:12 home/alex/.local/share/gsettings-data-convert

↘loads or adjusts the sound manager (could be Window Manager).
We find empty folders in /tmp/pulse-* which would have contained sound cache.

12 2012-08-23 01:32:13 home/alex/.pulse-cookie
13 2012-08-23 01:32:13 home/alex/.pulse/75859b163e0beed48d8819b700000002-device-volumes.tdb
14 2012-08-23 01:32:13 home/alex/.pulse/75859b163e0beed48d8819b700000002-card-database.tdb
15 2012-08-23 01:32:13 home/alex/.fontconfig/cabbd14511b9e8a55e92af97fb3a0461-le32d4.cache-3
16 2012-08-23 01:32:16 home/alex/.gconf/apps/%gconf.xml
17 2012-08-23 01:32:16 home/alex/.gconf/apps/nm-applet/%gconf.xml
18 2012-08-23 01:32:16 home/alex/.local/share/.converted-launchers

↘Suggestion we are in a graphical user environment

19 2012-08-23 01:32:16 home/alex/.cache/unity/migration_script.log

↘hud-usage-log.sqlite reveals no usage

20 2012-08-23 01:32:20 home/alex/.cache/indicator-appmenu/hud-usage-log.sqlite
21 2012-08-23 01:32:20 home/alex/.local/share/zeitgeist/fts.index/iamchert
22 2012-08-23 01:32:24 home/alex/.cache/unity-lens-video/videos.db

↘No Telepathy accounts configured

23 2012-08-23 01:32:29 home/alex/.mission-control/accounts/accounts.cfg
24 2012-08-23 01:33:15 home/alex/.cache/update-manager-core/meta-release-lts

↘non-standard terminal may collect application specific metadata; none found in these files

25 2012-08-23 01:33:26 home/alex/.gconf/apps/gnome-terminal/%gconf.xml
26 2012-08-23 01:33:26 home/alex/.gconf/apps/gnome-terminal/profiles/%gconf.xml

27 2012-08-23 01:33:26 home/alex/.gconf/apps/update-notifier/%gconf.xml
28 2012-08-23 01:33:26 home/alex/.gconf/apps/update-manager/%gconf.xml

↘single sign-on client GUI log empty; may investigate password database if found

29 2012-08-23 01:34:15 home/alex/.cache/sso/sso-client-gui.log
30 2012-08-23 01:35:14 home/alex/.gconf/apps/deja-dup/%gconf.xml
31 2012-08-23 01:35:14 home/alex/.gconf/apps/deja-dup/s3/%gconf.xml

↘uninteresting image

32 2012-08-23 01:36:26 home/alex/.cache/oneconf/75859b163e0beed48d8819b700000002/logo_75859b163e0beed48d8819b700000002.png
33 2012-08-23 01:46:36 home/alex/.fontconfig/3047814df9a2f067bd2d96a2b9c36e5a-le32d4.cache-3
34 2012-08-23 01:47:24 home/alex/.fontconfig/4794a0821666d79190d59a36cb4f44b5-le32d4.cache-3
35 2012-08-23 01:48:42 home/alex/.gnome2/keyrings/user.keystore.QNPIJW
36 2012-08-23 01:49:56 home/alex/.cache/ubuntuone/log/credentials.log

↘7 failed logins to Ubuntu One cloud (random intervals); may be a retrievable account username.

37 2012-08-23 01:54:26 home/alex/.local/share/gvfs-metadata/root
38 2012-08-23 01:55:27 home/alex/.config/Trolltech.conf

↘encrypted log contains strings: 'jour' 'P5q2' '/tmp/filemWgtgv' '/tmp/filez7R1Bg'

39 2012-08-23 01:57:18 home/alex/.local/share/gvfs-metadata/root-69945aef.log

↘uninteresting images (all checked)

40 2012-08-23 01:59:47 home/alex/.thumbnails/normal/80669861f9aa5aea3fe60cb442f453cc.png
.... *continued thumbnail access (*.thumbnails/normal/*)*
42 2012-08-23 01:59:47 home/alex/.thumbnails/normal/e7570bde072b0d684f4717fcfbe08d50.png
52 2012-08-23 01:59:47 home/alex/.thumbnails/normal/a3798128bdc1f80811d42f9504d8821a.png
53 2012-08-23 01:59:47 home/alex/.thumbnails/fail/gnome-thumbnail-factory/606a1c08d54b3ec958ba03c23bc6de3c.png
54 2012-08-23 01:59:47 home/alex/.thumbnails/normal/bf5ba75daabead8f88aac1db4c5fa124.png
55 2012-08-23 01:59:47 home/alex/.thumbnails/normal/beecf827de87042a2c714ac89e00e7ea.png

↘loads Mozilla Thunderbird mail client

56 2012-08-23 02:01:51 home/alex/.thunderbird/Crash Reports/InstallTime20120714011110
57 2012-08-23 02:01:51 home/alex/.thunderbird/h9jaeoos.default/secmod.db
58 2012-08-23 02:01:51 home/alex/.thunderbird/h9jaeoos.default/extensions.ini
59 2012-08-23 02:01:51 home/alex/.thunderbird/h9jaeoos.default/extensions.sqlite
60 2012-08-23 02:01:51 home/alex/.thunderbird/profiles.ini
61 2012-08-23 02:01:52 home/alex/.thunderbird/h9jaeoos.default/pluginreg.dat
62 2012-08-23 02:01:52 home/alex/.thunderbird/h9jaeoos.default/mailViews.dat
63 2012-08-23 02:01:53 home/alex/.thunderbird/h9jaeoos.default/cookies.sqlite
64 2012-08-23 02:01:53 home/alex/.thunderbird/h9jaeoos.default/chromeappsstore.sqlite
65 2012-08-23 02:02:03 home/alex/.thunderbird/h9jaeoos.default/webappsstore.sqlite
66 2012-08-23 02:02:53 home/alex/.cache/indicators/messages/seen-db.keyfile
67 2012-08-23 02:03:51 home/alex/.thunderbird/h9jaeoos.default/search.json

2012-08-22 02:04:41 Article-A1: instruction to recon the server. Hasn't seen message yet.

68 2012-08-23 02:04:55 home/alex/.thunderbird/h9jaeoos.default/Mail/Local Folders/Trash
69 2012-08-23 02:04:55 home/alex/.thunderbird/h9jaeoos.default/Mail/Local Folders/Unsent Messages

↘may have polled email server, or may be part of application startup process

70 2012-08-23 02:04:55 home/alex/.thunderbird/h9jaeoos.default/signons.sqlite
71 2012-08-23 02:04:58 home/alex/.gconf/desktop/%gconf.xml
72 2012-08-23 02:04:58 home/alex/.gconf/desktop/gnome/%gconf.xml
73 2012-08-23 02:04:58 home/alex/.gconf/desktop/gnome/url-handlers/%gconf.xml
74 2012-08-23 02:04:58 home/alex/.gconf/desktop/gnome/url-handlers/mailto/%gconf.xml
75 2012-08-23 02:04:58 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/msgFilterRules.dat
76 2012-08-23 02:04:58 home/alex/.local/share/mime/aliases
77 2012-08-23 02:04:58 home/alex/.local/share/mime/generic-icons
78 2012-08-23 02:04:58 home/alex/.local/share/mime/icons
79 2012-08-23 02:04:58 home/alex/.local/share/mime/subclasses
80 2012-08-23 02:04:58 home/alex/.local/share/mime/XMLnamespaces
81 2012-08-23 02:04:58 home/alex/.local/share/mime/magic
82 2012-08-23 02:04:58 home/alex/.local/share/mime/globs
83 2012-08-23 02:04:58 home/alex/.local/share/mime/globs2
84 2012-08-23 02:04:58 home/alex/.local/share/mime/treemagic
85 2012-08-23 02:04:58 home/alex/.local/share/mime/mime.cache
86 2012-08-23 02:04:58 home/alex/.local/share/mime/packages/user-extension-eml.xml
87 2012-08-23 02:04:58 home/alex/.local/share/mime/types
88 2012-08-23 02:04:58 home/alex/.local/share/mime/application/x-extension-eml.xml
89 2012-08-23 02:04:58 home/alex/.local/share/applications/mimeapps.list
90 2012-08-23 02:04:58 home/alex/.local/share/mime/version
91 2012-08-23 02:04:59 home/alex/.thunderbird/h9jaeoos.default/Cache/9/C0/EFAF1d01
92 2012-08-23 02:05:00 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/Sent.msf
93 2012-08-23 02:05:00 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/Drafts.msf
94 2012-08-23 02:05:00 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/Archives.msf
95 2012-08-23 02:05:01 home/alex/.pulse/75859b163e0beed48d8819b700000002-stream-volumes.tdb
96 2012-08-23 02:05:01 home/alex/.thunderbird/h9jaeoos.default/abook.mab

97 2012-08-23 02:05:09 home/alex/.thunderbird/h9jaeoos.default/urlclassifier.pset
98 2012-08-23 02:05:09 home/alex/.thunderbird/h9jaeoos.default/urlclassifier3.sqlite
↘remote IMAP account is loaded (Google inc.). Likely found message 1 at this point.
99 2012-08-23 02:05:21 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com.msf
100 2012-08-23 02:07:05 home/alex/.mozilla/firefox/Crash Reports/InstallTime20120713224758
101 2012-08-23 02:07:05 home/alex/.mozilla/firefox/profiles.ini
102 2012-08-23 02:07:06 home/alex/.mozilla/firefox/wbaoqfjsm.default/secmod.db
103 2012-08-23 02:07:06 home/alex/.mozilla/firefox/wbaoqjsm.default/search.json
104 2012-08-23 02:07:06 home/alex/.mozilla/firefox/wbaoqjsm.default/extensions.sqlite
105 2012-08-23 02:07:06 home/alex/.mozilla/firefox/wbaoqjsm.default/extensions.ini
106 2012-08-23 02:07:06 home/alex/.mozilla/firefox/wbaoqjsm.default/mimeTypes.rdf
107 2012-08-23 02:07:16 home/alex/.mozilla/firefox/wbaoqjsm.default/chromeappsstore.sqlite
108 2012-08-23 02:07:17 home/alex/.mozilla/firefox/wbaoqjsm.default/downloads.sqlite
109 2012-08-23 02:11:06 home/alex/.mozilla/firefox/wbaoqjsm.default/pluginreg.dat
110 2012-08-23 02:12:50 home/alex/.fontconfig/6d41288fd70b0be22e8c3a91e032eec0-le32d4.cache-3
111 2012-08-23 02:13:22 home/alex/.cache/oneconf/75859b163e0beed48d8819b700000002/host
112 2012-08-23 02:13:22
home/alex/.cache/oneconf/75859b163e0beed48d8819b700000002/package_list_75859b163e0beed48d8819b700000002
113 2012-08-23 02:18:34 home/alex/.cache/sso/sso-client.log
↘uninteresting bookmarks (standard distribution)
114 2012-08-23 02:29:09 home/alex/.mozilla/firefox/wbaoqjsm.default/bookmarkbackups/bookmarks-2012-08-23.json
115 2012-08-23 02:49:47 home/alex/.subversion/README.txt
↘suggests credentials may be stored in 'gnome-keyring' or 'kwallet'
116 2012-08-23 02:49:47 home/alex/.subversion/config
↘No servers or credentials configured in this file (below)
117 2012-08-23 02:49:47 home/alex/.subversion/servers
↘[Listing C: bash history:36] suggests an SVN repository checkout from https://svn.nmap.org/nmap to ~/Development/
118 2012-08-23 02:49:53 home/alex/Development/nmap/scripts/.svn/text-base/stun-version.nse.svn-base
.... *continued uninterrupted nmap download*
*(*home/alex/Development/nmap/*)*
Folders of interest: scripts, ndiff, nping, ncat, zenmap, OpenSSL, & many more. Cross references with Pepys server breach audit.
3852 2012-08-23 02:50:15 home/alex/Development/nmap/zenmap/zenmap
↘9 second gap suggests are large file or intermittent activity
3853 2012-08-23 02:50:24 home/alex/Development/nmap/mswin32/OpenSSL/include/openssl/.svn/text-base/des.h.svn-base
.... *continued uninterrupted nmap download*
*(*home/alex/Development/nmap/mswin32/*)*
4307 2012-08-23 02:50:28 home/alex/Development/nmap/mswin32/winpcap/vista/x86/Packet.dll
↘18 second gap suggests are large file or intermittent activity
4308 2012-08-23 02:50:46 home/alex/Development/nmap/.svn/text-base/protocols.cc.svn-base
.... *continued uninterrupted nmap download*
*(*home/alex/Development/nmap/*)*
4665 2012-08-23 02:50:47 home/alex/Development/nmap/libdnet-stripped/libdnet-stripped.vcxproj
↘23 second gap suggests are large file or intermittent activity [Listing C: bash history:39] suggests compilation began
4666 2012-08-23 02:51:10 home/alex/Development/nmap/libnetutil/Makefile
.... *continued nmap compliation*
4736 2012-08-23 02:53:07 home/alex/Development/nmap/zenmap/INSTALLED_FILES
4737 2012-08-25 02:47:27 home/alex/.cache/wallpaper/0_5_1360_768_792beab7550410d531e55f95b449f135
4738 2012-08-25 02:47:29 home/alex/.config/monitors.xml
↘Interspersed Mozilla Firefox and Thunderbird activity: suggests they are both loaded and being used simultaneously.
4739 2012-08-25 02:47:47 home/alex/.mozilla/firefox/wbaoqjsm.default/compatibility.ini
4740 2012-08-25 02:47:59 home/alex/.thunderbird/h9jaeoos.default/compatibility.ini
4741 2012-08-25 02:48:05 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/Templates.msf
4742 2012-08-25 02:48:07 home/alex/.gconf/system/%gconf.xml
4743 2012-08-25 02:48:07 home/alex/.gconf/system/http_proxy/%gconf.xml
4744 2012-08-25 02:50:00 home/alex/.thunderbird/h9jaeoos.default/addons.sqlite
4745 2012-08-25 02:51:48 home/alex/.mozilla/firefox/wbaoqjsm.default/addons.sqlite
4746 2012-08-25 02:52:01 home/alex/.thunderbird/h9jaeoos.default/Cache/4/54/A8679d01
4747 2012-08-25 02:52:01 home/alex/.thunderbird/h9jaeoos.default/blocklist.xml
4748 2012-08-25 02:52:01 home/alex/.thunderbird/h9jaeoos.default/permissions.sqlite
↘IMAP .msf box has different filename from line 99 (above), but there is only one mail account at present
4749 2012-08-25 02:53:00 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].msf
4750 2012-08-25 02:53:49 home/alex/.mozilla/firefox/wbaoqjsm.default/blocklist.xml
4751 2012-08-25 02:53:49 home/alex/.mozilla/firefox/wbaoqjsm.default/permissions.sqlite
↘'alex' accessed reconnaissance gathering documents for the last time
4752 2012-08-25 02:54:30 home/alex/hacks/user-list.txt
4753 2012-08-25 02:55:00 home/alex/hacks/pass-list.txt
↘checked (perhaps cleared) Trash box
4754 2012-08-25 02:55:22 home/alex/.thunderbird/h9jaeoos.default/Mail/Local Folders/Trash.msf

4755 2012-08-25 02:55:30 home/alex/.mozilla/firefox/wbaoqjsm.default/bookmarkbackups/bookmarks-2012-08-25.json
↘0 byte dump file
4756 2012-08-25 03:06:38 home/alex/.mozilla/firefox/wbaoqjsm.default/minidumps/186fbace-19e0-13bb-3fb75aeb-14362c37.dmp
↘cache indicates browsing of strongVPN.nl. Checking reviews
4757 2012-08-25 03:08:54 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/5/0E/02AAEd01
…. … … … … … … … … … *continued Mozilla Firefox (browsing) activity (*home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/*)*
4786 2012-08-25 03:09:04 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/7/BF/1EF50d01
4787 2012-08-25 17:46:31 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/F/15/4361Ed01
4788 2012-08-25 17:46:37 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Spam.msf
4789 2012-08-25 17:46:37 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Starred.msf
4790 2012-08-25 17:48:26 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/D/8A/41EA7d01
4791 2012-08-25 17:48:26 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/B/66/E3474d01
↘png image of nieuwsblad.be front page
4792 2012-08-25 17:48:27 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/B/B6/C8546d01
↘cache indicates continued browsing of nieuwsblad.be
Images include a Dutch Election 2012 advert, an owl, cyclists, cartoon of an American capturing a fugitive .. articles appear insignificant
…. … … … … … … … … … *continued Mozilla Firefox (browsing) activity (*home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/*)*
4834 2012-08-25 17:48:43 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/2/7A/3D526d01
4835 2012-08-25 17:53:26 home/alex/.thunderbird/h9jaeoos.default/places.sqlite
↘More images (appear to be a side product of browsing Nieuwsblad.be).
uploading some images.google.com matches to URL articles found in Mozilla Firefox cache.
4836 2012-08-25 17:53:42 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/0/63/528C7d01
…. … … … … … … … … … *continued Mozilla Firefox (browsing) activity (*home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/*)*
↘**png image of Nieuwsblad.be article found in Mozilla Firefox cache:**
http://www.nieuwsblad.be/article/detail.aspx?articleid=DMF20120825_057
Note how the RSS feed of recent news is 17.47;
aligning to our cache file's timestamp and our corresponding firefox browser history line [Appendix B: Browser history:8] **17:53:42)**
4857 2012-08-25 17:53:57 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/A/83/94C48d01
4858 2012-08-25 18:06:24 home/alex/.thunderbird/h9jaeoos.default/Mail/Local Folders/Unsent Messages.msf
↘Recent files listed in .local/share/recently-used.xbel
4859 2012-08-25 18:11:02 home/alex/.gconf/apps/gedit-2/%gconf.xml
4860 2012-08-25 18:11:02 home/alex/.gconf/apps/gedit-2/preferences/%gconf.xml
4861 2012-08-25 18:11:02 home/alex/.gconf/apps/gedit-2/preferences/ui/%gconf.xml
4862 2012-08-25 18:11:04 home/alex/.config/enchant/en_US.dic
4863 2012-08-25 18:11:04 home/alex/.config/enchant/en_US.exc
↘cache of MetaSploit "TCP Port Scanner" download page (gzipped js file and png images)
4864 2012-08-25 18:12:25 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/C/03/D6272d01
…. … … … … … … … … … *continued Mozilla Firefox (browsing) activity (*home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/*)*
4869 2012-08-25 18:13:25 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/8/E6/F3460d01
4870 2012-08-25 18:17:03 home/alex/.gconf/apps/nautilus/%gconf.xml
4871 2012-08-25 18:17:03 home/alex/.gconf/apps/nautilus/preferences/%gconf.xml
4872 2012-08-25 18:59:58 home/alex/.ssh/known_hosts
↘HTML browsable cache of the Pepys blog
4873 2012-08-25 19:00:59 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/A/0C/8D281d01
↘png image of the Pepys blog
4874 2012-08-25 19:01:00 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/4/20/8DB90d01
4875 2012-08-25 19:05:16 home/alex/hacks/user-list-ssh.txt
↘png image of the Pepys blog Wordpress dashboard
4876 2012-08-25 19:12:16 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/9/43/00F42d01
4877 2012-08-25 19:12:30 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/F/97/7D5B9d01
↘Javascript from Wordpress (likely dashboard)
4878 2012-08-25 19:12:31 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/A/84/5541Fd01
4879 2012-08-25 19:17:01 home/alex/hacks/user-list-wordpress.txt
4880 2012-08-25 19:17:41 home/alex/.msf4/logs/framework.log
↘A gzipped javascript file; source code contains "www.metasploit.com", "rapid7.com"
4881 2012-08-25 19:18:45 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/1/02/B2CBBd01
↘Metasploit issued commands log
4882 2012-08-25 19:24:33 home/alex/.msf4/history
↘more compressed javascript

4883 2012-08-25 19:26:08 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/E/F8/90817d01
.... *continued Mozilla Firefox (browsing) activity (*home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/*)*
↘png image of manpages.ubuntu.com's ("Hardy") MySQL manual page (preceeded by Google search "man mysql")
4895 2012-08-25 19:26:33 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/3/64/10592d01
4896 2012-08-25 19:38:50 home/alex/hacks/pepys.dyndns.org.reconn.txt~
4897 2012-08-25 19:40:34 home/alex/hacks/pepys.dyndns.org.reconn.txt
↘png image of MetaSploit "MySQL Login Utility" download page
4898 2012-08-25 19:42:28 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/2/EB/C9EBFd01
↘png image of MetaSploit "Brute Force and User Enumeration Utility" download page
4899 2012-08-25 19:42:30 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/2/12/20F99d01
↘png image of MetaSploit "SSH Login Check" download page
4900 2012-08-25 19:42:31 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/E/B6/316FDd01
4901 2012-08-25 19:42:46 home/alex/.cache/event-sound-cache.tdb.75859b163e0beed48d8819b700000002.i686-pc-linux-gnu
4902 2012-08-25 19:43:08 home/alex/.xsession-errors.old
4903 2012-08-25 20:23:09 home/alex/.dmrc
4904 2012-08-25 20:23:10 home/alex/.dbus/session-bus/75859b163e0beed48d8819b700000002-0
4905 2012-08-25 20:23:11 home/alex/.ICEauthority
4906 2012-08-25 20:23:15 home/alex/.gtk-bookmarks
4907 2012-08-25 20:23:19 home/alex/.pulse/75859b163e0beed48d8819b700000002-default-source
4908 2012-08-25 20:23:19 home/alex/.pulse/75859b163e0beed48d8819b700000002-default-sink
4909 2012-08-25 20:23:20 home/alex/.config/nautilus/desktop-metadata
↘No IM accounts configured
4910 2012-08-25 20:24:09 home/alex/.local/share/telepathy/mission-control/accounts-goa.cfg
4911 2012-08-25 20:24:13 home/alex/.local/share/zeitgeist/fts.index/flintlock
4912 2012-08-25 20:24:21 home/alex/.mozilla/firefox/wbaoqjsm.default/.parentlock
4913 2012-08-25 20:24:23 home/alex/.mozilla/firefox/wbaoqjsm.default/sessionstore.bak
4914 2012-08-25 20:24:24 home/alex/.mozilla/firefox/wbaoqjsm.default/urlclassifierkey3.txt
4915 2012-08-25 20:28:45 home/alex/hacks/john/mbox
4916 2012-08-25 20:30:03 home/alex/hacks/samuel/samuel
↘analyzed with SQLite manager; list of clicked applications & files retrieved
4917 2012-08-25 20:33:31 home/alex/.local/share/zeitgeist/activity.sqlite
4918 2012-08-25 20:33:36 home/alex/.gconf/apps/gedit-2/preferences/ui/statusbar/%gconf.xml
4919 2012-08-25 20:33:36 home/alex/.gconf/apps/gedit-2/plugins/%gconf.xml
4920 2012-08-25 20:41:27 home/alex/hacks/mail-thread.txt~
4921 2012-08-25 20:48:15 home/alex/hacks/mail-thread.txt
4922 2012-08-25 20:48:25 home/alex/.config/gedit/accels
4923 2012-08-25 20:48:25 home/alex/.config/dconf/user
↘Picture of a group of rowers in Ghent, Belgium. uploading the images.google.com matches to:
http://www.nieuwsblad.be/article/detail.aspx?articleid=BLRTO_20120825_005
4924 2012-08-25 20:49:15 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/1/17/C80B2d01
↘Picture of footballers. uploading the images.google.com matches image to source websites. Appears to be the "Belgian ('Jupiler') Pro League"
4925 2012-08-25 20:49:15 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/2/C3/FB9A5d01
↘Picture of a crime scene on 53rd street NYC, USA. No online matches found
4926 2012-08-25 20:49:15 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/D/3D/FAD6Ed01
↘Picture from http://www.nieuwsblad.be/article/detail.aspx?articleid=DMF20120825_057
4927 2012-08-25 20:49:16 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/7/FE/19A31d01
↘Picture advert for XS4all.nl internet, email and hosting provider..request additional records (e.g. address, geolocation, additional browsing history).
4928 2012-08-25 20:49:22 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/1/AD/054E8d01
4929 2012-08-25 20:49:22 home/alex/.local/share/gvfs-metadata/home
4930 2012-08-25 20:49:22 home/alex/.local/share/gvfs-metadata/home-609e576e.log
4931 2012-08-25 20:49:37 home/alex/.thunderbird/h9jaeoos.default/.parentlock
4932 2012-08-25 20:49:43 home/alex/.thunderbird/h9jaeoos.default/Cache/8/F1/2FAB1d01
4933 2012-08-25 20:49:58 home/alex/.thunderbird/h9jaeoos.default/content-prefs.sqlite
4934 2012-08-25 20:59:59 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Drafts
4935 2012-08-25 21:02:56 home/alex/.thunderbird/h9jaeoos.default/history.mab
2012-08-25 15:02:56 Article-A2: Alexandrine responds with a mail thread
4936 2012-08-25 21:02:59 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Sent Mail
4937 2012-08-25 21:03:56 home/alex/.thunderbird/h9jaeoos.default/startupCache/startupCache.4.little
↘javascript (for configuring cookies and themes) and picture advertisements
4938 2012-08-25 21:04:24 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/C/73/67DDAd01
.... *continued Mozilla Firefox (browsing) activity (*home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/*)*
↘png image of nieuwsblad.be website (with one of cached adverts above embedded)
4943 2012-08-25 21:15:58 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/8/2D/45EC8d01
2012-08-25 15:16:27 Article-A3: suggestion to deface the website

4944 2012-08-25 21:16:31 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/INBOX
4945 2012-08-25 21:17:06 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/All Mail
4946 2012-08-25 21:17:08 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Important
4947 2012-08-25 21:17:10 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Trash
4948 2012-08-25 21:17:18 home/alex/Downloads/200px-Prinsenvlag.svg.png
4949 2012-08-25 21:17:27 home/alex/.thunderbird/h9jaeoos.default/global-messages-db.sqlite
4950 2012-08-25 21:17:27 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/All Mail.msf
4951 2012-08-25 21:17:27 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Sent Mail.msf
4952 2012-08-25 21:17:51 home/alex/.mozilla/firefox/wbaoqjsm.default/webappsstore.sqlite
4953 2012-08-25 21:19:17 home/alex/.mozilla/firefox/wbaoqjsm.default/signons.sqlite
4954 2012-08-25 21:19:40 home/alex/.thunderbird/h9jaeoos.default/session.json
4955 2012-08-25 21:21:13 home/alex/.mozilla/firefox/wbaoqjsm.default/content-prefs.sqlite
↘png image of the Pepys blog page/post editing panel
4956 2012-08-25 21:22:17 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/B/90/B8506d01
↘png image of the Pepys blog profile editing panel
4957 2012-08-25 21:22:30 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/7/72/341C5d01
↘png image of the Pepys blog users and associated URLs (shows Wordpress user "Alex", URL "alexking.org")
4958 2012-08-25 21:22:32 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/A/F7/604C7d01
↘analyzed with a gnome-activity-browser (see GUI history: Zeitgeist and gedit (GTK+))
4959 2012-08-25 21:24:15 home/alex/.local/share/zeitgeist/fts.index/record.baseB
4960 2012-08-25 21:24:15 home/alex/.local/share/zeitgeist/fts.index/termlist.baseB
4961 2012-08-25 21:24:15 home/alex/.local/share/zeitgeist/fts.index/position.baseA
4962 2012-08-25 21:24:15 home/alex/.local/share/zeitgeist/fts.index/postlist.baseB
4963 2012-08-25 21:25:16 home/alex/.gconf/apps/gnome-terminal/profiles/Default/%gconf.xml
4964 2012-08-25 21:27:11 home/alex/.mozilla/firefox/wbaoqjsm.default/urlclassifier3.sqlite
4965 2012-08-25 21:27:14 home/alex/.mozilla/firefox/wbaoqjsm.default/urlclassifier.pset
4966 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/record.baseA
4967 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/termlist.baseA
4968 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/position.baseB
4969 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/postlist.baseA
4970 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/record.DB
4971 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/position.DB
4972 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/postlist.DB
4973 2012-08-25 21:28:05 home/alex/.local/share/zeitgeist/fts.index/termlist.DB
↘png image of the Pepys blog page/post editing panel
4974 2012-08-25 21:29:37 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/5/49/00AB9d01
↘png image matches 200px-Prinsenvlag.svg.png
4975 2012-08-25 21:31:32 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/3/60/9A61Ed01
↘png image of the Pepys blog Wordpress dashboard
4976 2012-08-25 21:31:48 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/5/57/6E44Ed01
↘png image of the Pepys blog frontpage
4977 2012-08-25 21:32:17 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/0/9B/014EBd01
↘png image of the Pepys blog posts editing panel
4978 2012-08-25 21:32:19 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/1/D6/36412d01
↘png image of the Pepys blog page/post editing panel with text entered
↘"read about .. how our nation should support Republiek der Zeven Verenigde Nederlanden!"
4979 2012-08-25 21:32:26 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/4/8C/7EC88d01
4980 2012-08-25 21:32:51 home/alex/.mozilla/firefox/wbaoqjsm.default/formhistory.sqlite
↘png image of the Pepys blog page management panel
Page ID: 2, title: About, owner:Samuel, edited 2012-08-25 8:33pm (timezone may be inferred from host)
We note that this time (8.33pm) is NOT UTC or CEST, BECAUSE: `ls -lt --full-time` or `ls -ltu --full-time` *both* report 19.33:55 UTC (definitely, definitely, definitely) for this file.
The Wordpress site must have been set to BST (UTC+1).
4981 2012-08-25 21:32:55 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/3/84/38ABFd01
4982 2012-08-25 21:32:55 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/E/25/FA1B1d01
4983 2012-08-25 21:33:11 home/alex/.mozilla/firefox/wbaoqjsm.default/startupCache/startupCache.4.little
4984 2012-08-25 21:34:17 home/alex/.bash_history
↘HTML browsable cache of the Pepys blog archive
4985 2012-08-25 21:34:23 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/D/66/17AE7d01
↘png image of the Pepys blog frontpage
4986 2012-08-25 21:34:35 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/1/9C/31CF9d01
4987 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/_CACHE_001_
↘saved Mozilla Firefox website login credentials
4988 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/key3.db
4989 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/_CACHE_003_
4990 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/sessionstore.js
4991 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/prefs.js

4992 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/cert8.db
4993 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/_CACHE_MAP_
4994 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/Cache/_CACHE_002_
↘A trove of Mozilla Firefox profile data from which we exported [Appendix B: Browser history]
4995 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/places.sqlite
4996 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/cookies.sqlite
4997 2012-08-25 21:34:41 home/alex/.mozilla/firefox/wbaoqjsm.default/localstore.rdf
4998 2012-08-25 21:34:45 home/alex/.thunderbird/h9jaeoos.default/folderTree.json
4999 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/key3.db
5000 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/Cache/_CACHE_003_
5001 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/Cache/_CACHE_002_
5002 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/Cache/_CACHE_001_
5003 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/cert8.db
5004 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/prefs.js
5005 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/Cache/_CACHE_MAP_
5006 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/virtualFolders.dat
5007 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/localstore.rdf
5008 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Drafts.msf
5009 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Trash.msf
5010 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/[Gmail].sbd/Important.msf
5011 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/ImapMail/imap.googlemail.com/INBOX.msf
5012 2012-08-25 21:34:46 home/alex/.thunderbird/h9jaeoos.default/panacea.dat
5013 2012-08-25 21:34:59 home/alex/.cache/dconf/user
5014 2012-08-25 21:35:00 home/alex/.Xauthority
↘contains Zeitgeist / gedit 2 GUI editor recent files (see GUI history: Zeitgeist and gedit (GTK+))
5015 2012-08-25 21:35:00 home/alex/.local/share/recently-used.xbel
5016 2012-08-25 21:35:00 home/alex/.cache/indicator-applet-complete.log
5017 2012-08-25 21:35:01 home/alex/.xsession-errors
5018 2012-08-25 21:35:02 home/alex/.local/share/zeitgeist/activity.sqlite-wal
5019 2012-08-25 21:35:02 home/alex/.local/share/zeitgeist/activity.sqlite-shm

END OF TIMELINE

We though at length about commenting the file modification time vs. the file access time [Notes: Filesystem listings: export ID 5 vs 6]. In this case, where reviewing the data seemed to imply that actions were being performed for the first time and one time only, listing the modification times showed us an order in which things were done; i.e. To create a file, and then browse a page for the first time, and then to download an image, etc. When creating a timeline, we want the **first instance** an attachment being saved (we can obtain the time of upload from elsewhere). It is a careful decision depending on the case, and really requires both exports; you will see that we also reference the access times [Notes: Filesystem listings: export ID 6] in the next section.

Interactive shell histories

notes: shell logs

We have complete lists of concrete access[4] and modification[5] timestamps from our home directory, and entire filesystem[6,7].
When predicting the order of creation we have preferred modification times; when predicting the order of subsequent activity (after files have already been created) we have referred to access times[8].

We now find the histories of two interactive shells - we describe the chronological activity that took place, and attempt to timestamp the commands from our research.

Metasploit history

We saw plenty of reference alluding to the download of 'Metasploit' utilities. We also find additional timestamps which we wish to comprehend and attribute to activity (`home/alex/.msf4/logs/framework.log`).

Now that we have a complete list of times from 3 different sources (browser URL history, filesystem modification (*often creation*) & access times, Metasploit error log) we can go through the Metasploit shell history (`home/alex/.msf4/history`) attempting to verify (and assign times to) what took place.

---

[4] http://users.ox.ac.uk/~kell3138/FOR/find_alex_access_times.txt
[5] http://users.ox.ac.uk/~kell3138/FOR/find_alex_modification_times.txt
[6] http://users.ox.ac.uk/~kell3138/FOR/all_access_times.txt
[7] http://users.ox.ac.uk/~kell3138/FOR/all_modification_times.txt
[8] where both modification and access times are the same the file has been created or altered but never subsequently used.

Listing B: Metasploit history

Chronological command list (reading down)	Comment	Estimated time (Mozilla Firefox places.sqlite)	Estimated time (browser cache file modification time)	Framework error log * home/alex/.msf4/logs/framework.log
		|-high correlation-|	
exit				
	(New session)			
use auxiliary/scanner/portscan/tcp	activate 'TCP Port Scanner' utility	25/08/2012 18:12:25	25/08/2012 18:12:25	08/25/2012 18:07:57
set RHOSTS pepys.dyndns.org	Set target			
set PORTS 1-1024	set port range			
info	print module information	about 1 minute	about 1 minute	
exploit	launch exploit			
use auxiliary/scanner/ssh/ssh_login	activate 'SSH Login Check' utility	25/08/2012 18:13	25/08/2012 18:13:39	
info	print module information			
set RHOSTS pepys.dyndns.org	Set target			
set BLANK_PASSWORDS false	Don't try "blank" password			
set PASS_FILE ~/hacks/pass-list.txt	Use local passwords list			
set USER_FILE ~/hacks/user-list.txt	Use local usernames list			
info	print module information			
exploit	launch exploit			about 55 minutes
info	print module information			
set PASS_FILE /home/alex/hacks/pass-list.txt	Use local passwords list	about 50 minutes		
set USER_FILE /home/alex/hacks/user-list.txt	Use local usernames list			
info	print module information			
exploit	launch exploit			
creds	list gathered credentials			
services	list services (e.g. OS, HTTP server)			
services -u	Only show running services			
vulns	list matched vulnerabilities		about 65 minutes	
exit	*(exit session)*			
exit -y	*(exit session - no prompt)*			08/25/2012 19:01:12
exit	*(exit session)*			about 5 minutes
info	print module information			

use auxiliary/scanner/http/wordpress_login_enum	activate 'Wordpress Brute Force and User Enumeration' utility
info	print module information
set BLANK_PASSWORDS false	Don't try "blank" password
set RHOSTS pepys.dyndns.org	Set target
set USER_FILE /home/alex/hacks/user-list-ssh.txt	Use local usernames list
set PASS_FILE /home/alex/hacks/pass-list.txt	Use local passwords list
info	print module information
exploit	launch exploit
creds	list gathered credentials
services	list services (e.g. OS, HTTP server)
services -u	Only show running services
vulns	list matched vulnerabilities
exit	*(exit session)*
creds	list gathered credentials
use auxiliary/scanner/mysql/mysql_login	activate 'MySQL Login' utility
info	print module information
set BLANK_PASSWORDS false	Don't try "blank" password
set RHOSTS pepys.dyndns.org	Set target
set PASS_FILE /home/alex/hacks/pass-list.txt	Use local passwords list
set USER_FILE /home/alex/hacks/user-list-wordpress.txt	Use local usernames list
info	print module information
use auxiliary/scanner/portscan/tcp	activate 'TCP Port Scanner' utility
set PORTS 3306	set port
set RHOSTS pepys.dyndns.org	Set target
info	print module information
exploit	launch exploit
services	list services (e.g. OS, HTTP server)
services -u	Only show running services
exit	*(exit session)*

25/08/2012 19:04:03

about 15 minutes

25/08/2012 19:18:44

08/25/2012 19:05:54

about 10 minutes

08/25/2012 19:17:41

25/08/2012 19:18:44

* Having run '`/NRO-image/metasploit-framework/msfconsole`' ourselves we find that it is the msf4 initialization process that creates this error (not the `'exit'`, nor the loading of modules). Since we presume that `msf4` was started *after* an `'exit'` command (a potentially long interval) and before the next command (likely a short interval) we have put the approximation next to the command which *follows* 'exit'.

bash history

We can also go through the bash history (`home/alex/.bashrc`) attempting to verify (and assign times to) what took place.

Listing C: bash history

Chronological command list (reading down)	comment
df -h	*(New session)* check partition(s) disk space
sudo apt-get remove libreoffice	Remove "Libre Office"
sudo apt-get autoremove	
sudo apt-get clean	remove package cache
df -h	check partition(s) disk space
aptcache show libreoffice	
apt-cache show libreoffice	
sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade	Upgrade distribution (versions in grub.cfg)
sudo apt-get remove libreoffice	Remove "Libre Office" after upgrade
sudo apt-get autoremove	
sudo apt-get clean	
df -h	
exit	exit session
df -h	*(New session)* check partition(s) disk space
exit	
sudo apt-get install gnome-panel	*(New session)* install "GNOME Panel"
df -h	check partition(s) disk space
exit	exit session
df -h	check partition(s) disk space
exxit	
exit	exit session
su postgres	run "PostgreSQL" database (as admin)
exit	exit session
echo export MSF_DATABASE_CONFIG= /opt/metasploit-framework/database.yml >> /etc/profile	Add Metasploit environment variable to /etc/profile
exit	exit session
sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade	Upgrade distribution (versions in grub.cfg)
df -h	
sudo apt-get install build-essential libreadline-dev libssl-dev libpq5 libpq-dev libreadline5 libsqlite3-dev libpcap-dev subversion openjdk-7-jre git-core autoconf postgresql pgadmin3 curl zlib1g-dev libxml2-dev libxslt1-dev vncviewer libyaml-dev ruby1.9.3 **/var/log/apt/history.log (line 41):** Start-Date: 2012-08-23 02:12:18 CEST Commandline: apt-get install build-essential libreadline-dev libssl-dev libpq5 libpq-dev libreadline5 libsqlite3-dev libpcap-dev subversion openjdk-7-jre git-core autoconf postgresql pgadmin3 curl zlib1g-dev libxml2-dev libxslt1-dev vncviewer libyaml-dev ruby1.9.3	
df -h	check partition(s) disk space
sudo apt-get clean	remove package cache
df -h	check partition(s) disk space

sudo gem install wirble msgpack sqlite3 pg activerecord nokogiri	install Ruby gems (packages)
cd mkdir ~/Development	
mkdir ~/Development	Create home/alex/Development/
cd Development/	Enter /home/alex/Development/
svn co https://svn.nmap.org/nmap	download nmap tool
df -h	check partition(s) disk space
cd nmap	Enter /home/alex/Development/nmap/
./configure	configure nmap for compilation
make	compile nmap
sudo make install	install nmap
make clean	clean redundant compilation files
df -h	check partition(s) disk space
sudo -s	enter admin shell (a.k.a. 'sudo su')
cd /opt	Enter /opt/
sudo svn co https://www.metasploit.com/svn/framework3/trunk metasploit-framework **2012-08-23 02:55:22 CEST** the timestamp of the first file to be downloaded	download Metasploit tools
cd metasploit-framework/	Enter /opt/metasploit-framework
df -h	check partition(s) disk space
sudo bash -c 'for MSF in $(ls msf*); do ln -s /opt/metasploit-framework/$MSF /usr/local/bin/$MSF;done'	symbolic link all files beginning with "msf" to /usr/local/bin/
sudo ln -s /opt/metasploit-framework/armitage /usr/local/bin/armitage	symbolic link armitage GUI to /usr/local/bin
sudo nano /opt/metasploit-framework/database.yml	Create/edit database configuration
sudo echo export MSF_DATABASE_CONFIG= /opt/metasploit-framework/database.yml >> /etc/profile	Add Metasploit environment variable to /etc/profile
sudo -s	enter admin shell (a.k.a. 'sudo su')
source /etc/profile	Reload /etc/profile
cd /opt/metasploit-framework/external/pcaprub	Enter ./external/pcaprub
sudo ruby extconf.rb && sudo make && sudo make install **metasploit-framework/external/pcaprub/Makefile:** modified: 2012-08-23 03:00:59 CEST, accessed: 2012-08-23 03:00:59 CEST This file was created/accessed once; if .bash_history is chronological then all previous commands took place before this date.	compile and make a Metasploit portscanning module dependency (http://www.darkoperator.com/installing-metasploit-in-ubunt/)
df -h	check partition(s) disk space

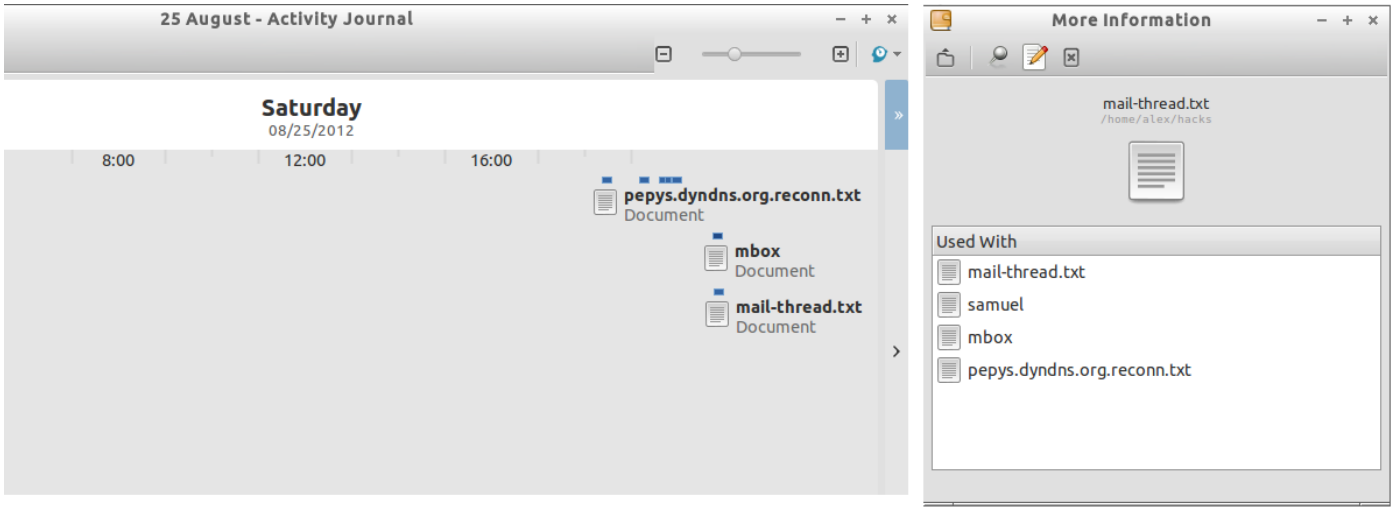
Our various timestamp sources line up to suggest that Metasploit was downloaded at 2012-08-23 02:55:22 CEST, and installed at about 2012-08-23 03:00:59 CEST.

From this we assert that the relevant bash history found is almost entirely regarding the installation of Metasploit, which was installed (`/opt/metasploit-framework`) the day following the first request from "Elizabeth Queen Of Bohemia" [Appendix A: Article-A1].

GUI history: Zeitgeist and gedit (GTK+) *notes: Zeitgeist and gedit (GTK+)*

When reviewing Notes: Filesystem listings: export ID 1 & 2 we saw 'Zeitgeist Activity Journal' was installed.

The 'alex 'user directory holds a Zeitgeist activity folder (`.local/share/zeitgeist`) containing activity databases, some of which are encrypted. Reloading this folder into our host system allows us view the files that were opened via the GUI:



Gnome activity Journal after importing Alex's Zeitgeist profile

Note how this application appears to match our filesystem times very accurately indeed!

gedit revisited

We inspected a GTK+ application history file (`local/share/recently-used.xbel`) which is used by 'gedit'.

In Listing A: Alex's timeline:4859-4863 we showed evidence that this text editor was loaded; since 'gedit' is the default GTK+ text editor, we make the connection that 'gedit' was used to used to graphically read or edit the following 4 text files:

"`home/alex/hacks/pepys.dyndns.org.reconn.txt`"
"`home/alex/hacks/john/mbox`"
"`home/alex/hacks/mail-thread.txt`"
"`home/alex/hacks/samuel/samuel`"

"`home/alex/Downloads/200px-Prinsenvlag.svg.png`" is also mentioned, but we believe this was logged to provide functionality for the "*Recently Used*" file system browser dialogue (a component of all GTK+ applications), and not necessarily opened by 'gedit'. Indeed, we ran numerous steganography checks and interrogations (including reading through the file [Notes: Applications of interest: internet browser: steganography] but found no trace of a hidden message.

The file lists legitimate timestamps encoded in the ISO 8601 format[9] (the original output can be seen in [Notes: Applications of interest: Zeitgeist and gedit (GTK+)]); UTC is a common default for applications where POSIX timestamps (seconds from 1970-01-01) are not given.

So checking `recently-used.xbel`'s UTC timestamps against our filesystem timestamp listings:

`home/alex/hacks/pepys.dyndns.org.reconn.txt" modified="2012-08-25T17:40:34Z"`
Filesystem modified times Listing A: Alex's timeline report 19:40.34 CEST (a.k.a. 17:40:34 UTC – exact match).

`home/alex/Downloads/200px-Prinsenvlag.svg.png" added="2012-08-25T19:17:17Z"`
Filesystem modified times Listing A: Alex's timeline reports UTC 20:17:18 (a.k.a. 19:17:17 UTC). 1 second out..perhaps there was a Zeitgeist or user delay when saving/opening the file.

From the exact match of "`pepys.dyndns.org.reconn.txt`" with the present filesystem state we might infer that the last modification was performed by a GTK+ application that updates 'recently-used.xbel' (e.g. '`gedit`').

---

[9] "*Example: 1385364167 (ISO 8601:2013-11-25T07:22:47Z)*",
Unix time, Wikipedia.org, http://en.wikipedia.org/wiki/Unix_time

"*Combined date and time in UTC: 2013-11-24T19:01Z*"
"<time>Z..If the time is in UTC, add a Z directly after the time without a space. Z is the zone designator for the zero UTC offset"
IS8601..Representation of dates and times (1988), Wikipedia.org, http://en.wikipedia.org/wiki/ISO_8601

Opinions

Following our investigation we now present the following piece to answer our initial questions.

The laptop was using a popular end user version of the Linux operating system.

It was geared towards general purpose home use (there was a full GUI environment).
Comparison with our own standard distribution showed little significant difference in the installation.

However, the office suite had been removed, and the additional security penetration software was downloaded after commands had been issued from the head of the organization. These tools included nmap[10] & Metasploit[11]. The latter's utilities were used for scanning the target, utilizing username and password lists found in the suspect's account home directory. The suspect was found to be communicating with the Queen of Bohemia using a local email client present on the machine. The messages and their headers were found to be intact.

Evidence strongly suggests that on Saturday August 25th CEST the suspect attacked the server. The correlation of browser logs, filesystem modification times, cached content and command history lists, as well as text documents found in the suspect's account containing valid credential and email content belonging to the victim, have all been retrieved. The email content was also emailed to the Queen.

In addition to circumstantial evidence (the presence of the offending image that was uploaded to Mr. Pepys' server in the suspect's inbox and account directory), we also have cached photographic evidence of the text being inserted into the target's website, as well as logged browser URLs that implicate the suspect to the scene.

We have recovered three IP addresses. We note that the header of a message from Alexandrine De Rye states her originating IP address as "172.16.77.129","213.179.209.92" (the latter of which is in Amsterdam, Netherlands). Our cache information (specifically the evidence of the recent newsfeed dates on http://www.nieuwsblad.be correlating perfectly to a CEST timezone) seems to support that she was located near here. However, we also noted the presence of a stored SSH server key (96.234.173.244) that matches the originating IP address of messages from Elizabeth Queen of Bohemia (Maryland, USA); suggesting that the suspect did connect to the Queen's computer via this protocol. Since the Metasploit history file indicates that all commands were executed locally, we are inclined to believe that 213.179.209.92 was where the attack was launched from. That said, from what we have researched of 'hacker practises', we might also believe that the SSH server was a second location from where an attack could have been launched.

Conclusion

Based on the evidence found we believe that Alexandrine de Rye, Countess of Thurn and Taxis, was the perpetrator of the attack on Mr. Samuel Pepys' blog, pepys.dyndns.org on the 25th of August 2012.

Recommendations

To prevent the success of such attacks in future, we recommend:

- Using a secure (lengthy and random character) password that cannot be guessed by a wordlist or by known contacts;
- Perform IP restrictions on the server, such that 3 failed login attempts will ban the IP address (preventing 'bruteforce' attacks);
- Always use the latest version of the available software, upgrading as part of a scheduled routine;
- Be very aware of social engineering infiltrations (people trying to gain information about family and friends).
- The rule of thumb in government institutions is ideally to "*trust no one*".

[10] Nmap Free Security Scanner For Network Exploration Audits, G. Lyon, http://nmap.org/
Google search engine returns a different title for this website: "Nmap Free Security Scanner For Network Exploration & Hacking"

[11] Metasploit: Penetration Testing Software, H. D. Moore, Rapid 7, www.metasploit.com/

Custody form	Case Control number 2013 10
Gold copy item: DVD1	
Items: DVD holing copy of Alexandrine De Rye laptop (partition only)	
Description: Hard Disk Data Acquisition	
Contents:	
Assignment\md5.derye.sda1.dd.txt	1,284,123,254 bytes
Assignment \derye.sda1.dd.zip	403 bytes
Assignment\ pepys.dyndns.org.etc.ssh.ssh_host_rsa_key.pub.txt	60 bytes
Assigned Investigator: M. PANT	ID: kell3138

Investigation protocol of a Digital forensic analyst **Date: 18th October 2013**

1) Insert Gold Copy DVD into host machine

2) Unzip compressed derye.sda1.dd.zip to local directory (`/home/forensic/case-FOR`)
[Notes: Preliminary set up on host machine]

3) Confirm integrity against hash file
[Notes: Preliminary set up on host machine]

4) Mount image
[Notes: Preliminary set up on host machine]

5) Investigate partitions
[Notes: Extra: slackspace and unallocated sectors with Autopsy]

6) Identify operating system (so we can download the original distribution's hash tables)
[Notes: Environment reconnaissance: operating system]

7) Identify local time zone
[Notes: Environment reconnaissance: clock calibration]

8) Identify active users (focus on suspect if appropriate)
[Notes: Environment reconnaissance: users]

9) Search for emails
[Notes: Applications of interest: email client]

10) Search browser history
[Notes: Applications of interest: internet browser]

11) Search for PGP/SSH keys and other applications
[Notes: Applications of interest: SSH]
[Notes: Applications of interest: keyrings]
[Notes: Applications of interest: instant messenger]

12) Retrieve filesystem access and modification listings
[Notes: Filesystem listings: export ID 1-6]

13) Catalogue filesystem with sorter (Autopsy)
[Extra: Generate a sorter timeline]

14) Search for deleted files and slackspace
[Extra: Slackspace and unallocated sectors with Autopsy]

15) Check for any concealed suspicious files with a virus scanner
[Extra: Concealed weapons]

Notes

Preliminary set up on host machine

We have an image of laptop partition on a CD.
We set it up on a host machine (a sandboxed virtual machine), where:

- The hostname is **i** for investigator
- The terminal prompt is in the format **<user>@<host>:<path of working directory> $**

The **$** changes to # if the account is an administrator. Command output has been included where relevant.

Process

Started my Investigator host virtual machine VMware Workstation (http://www.vmware.com/uk/products/workstation/)

Took a snapshot of the initial machine state (labelled '*initial boot*')

Loaded provided CDROM into the virtual machine

Copied provided image (`ASSIGNMENT/derye.sda1.dd.zip`) to standard user **forensic** (non-elevated permissions) home directory (`/home/forensic/case-FOR/`) via graphical user interface

1: Unzipped image (now non read-only) and mounted as read-only to host system via terminal:

```
forensic@i:/home/forensic$ cd case-FOR
forensic@i:/home/forensic/case-FOR# unzip derye.sda1.dd.zip
forensic@i:/home/forensic/case-FOR# sudo mkdir /mnt/derye_sda1
forensic@i:/home/forensic/case-FOR# sudo mount -o ro,noexec,loop derye.sda1.dd /mnt/derye_sda1
```

A shortcut was made to the mounted image (for readability):

```
forensic@i:/home/forensic/case-FOR# ln –s /mnt/derye_sda1 /RO-image
```

Confirmed MD5 sum with provided `md5.derye.sda1.dd.txt` (identical match confirmed):

```
forensic@i:/home/forensic/case-FOR# md5sum derye.sda1.dd 45e539b186b3e58d1654bf665d898f5e
```

Environment reconnaissance: operating system

We wish deduce the operating system so we may obtain a vanilla copy and run hash check comparisons if required.

```
forensic@i:/home/forensic$ cd /RO-image
forensic@i:/RO-image# ls proc
```

Unfortunately `/proc` is empty as that can feature OS information, as the command **uname** is a system call that gets its information from probing the kernel (files: `/proc/sys/kernel/{ostype, hostname, osrelease, version, domainname}` http://superuser.com/questions/509761/where-is-the-information-retreived-by-uname-stored)
The distribution name might be stored in the `/boot` or `/grub` directories. If `grub` prompts the user for the operating system at its boot menu we can check `boot/grub/grub.cfg` to find the operating system and hopefully the version.

```
forensic@i:/RO-image# less boot/grub/grub.cfg
..
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
```

..which is reassuring. More importantly it states the OS version as:

```
'Ubuntu, with Linux 3.2.0-29-generic-pae'
 insmod ext2              # filesystem type
```

```
 set root='(hd0,msdos1)'  # first harddrive partition - we can use these fields when looking
for slackspace later
```

We note that history of distribution installations:

```
submenu "Previous Linux versions" { menuentry 'Ubuntu, with Linux 3.2.0-23-generic-pae'
```

and the lines:

```
### BEGIN /etc/grub.d/40_custom ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment...if [ -f  $prefix/custom.cfg ]; then source
$prefix/custom.cfg;
..
### END /etc/grub.d/41_custom ###
```

..of which there are no additional menu entries entered or additional config files to examine.

```
forensic@i:/RO-image# cat etc/issue   # http://www.ubuntuka.com/how-to-find-out-ubuntu-version
 Ubuntu 12.04.1 LTS \n \l
forensic@i:/RO-image# cat etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04.1 LTS"

forensic@i:/RO-image# cat etc/debian_version
wheezy/sid
# debian 'sid' is the unstable branch [http://www.debian.org/releases/sid/]..exploitable bugs?
```

Now we have the distribution version we can go to the Ubuntu website (or check with NIST) to see if we can source MD5/SHA1 hashes for this installation (and/or the previous version). This will allows us to efficiently dismiss vanilla system files and focus exclusively on what the attacker has touched.

We check the distribution against the Ubuntu file folder structure[1] — we need to be thorough with all in case they are hiding directories in system folders, so distribution release hash checks are the most reliable and automatic approach.

Environment reconnaissance: hostname

```
forensic@i:/RO-image# less /etc/hostname # 1 line: the name of the machine is 'hacker-box'
forensic@i:/RO-image# nl /etc/hosts     # line 2: confirmation
```

Environment reconnaissance: timezone

Let us check the suspect's plain text `timezone` file.

```
forensic@i:/RO-image# cat etc/timezone
Europe/Brussels
```

`etc/localtime` is a timezone file; `zdump` can print the local time of the suspect's `etc/localtime` file.
To calculate the offset with ours, we first want to confirm our host machine's timezone settings:

```
forensic@i:/RO-image# ls –l /etc/localtime    # check the timezone on our host machine
lrwxrwxrwx 1 root root 23 Oct 28 05:53 /etc/localtime -> /usr/share/zoneinfo/UTC
# host machine timezone is set to UTC
forensic@i:/RO-image# dpkg-reconfigure tzdata # select UTC via Debian OS tool to be sure
forensic@i:/RO-image# ntpdate pool.ntp.org # synchronize our clock to NTP within milliseconds
02 Nov 08:27:58 ntpdate[16560]: adjust time server 193.47.164.28 offset 0.058903 sec
forensic@i:/RO-image# date
Sat Nov 02 08:28:07 UTC 2013     # confirm online; yes our clock is on UTC
```

[1] Ubuntu directory structure, R. Cotescu, http://www.linuxidentity.com/us/down/articles/LSK_U_9.04_directories_US.pdf

We are now on UTC. Now we want to check the suspect's timezone file.

We know that when we run `zdump <timezone file>` it will print that timezone file's localtime. E.g.:

```
forensic@i:/RO-image# zdump /usr/share/zoneinfo/EST /etc/timezone
/usr/share/zoneinfo/EST  Sat Nov 02 03:41:36 2013 EST # local time from our /usr/share/zone/EST
/etc/localtime           Sat Nov 02 08:41:36 2013 UTC # time from our /etc/localtime link
```

So you can see how probing the timezone file with `zdump` should tell us the offset between the configured `timezone` file (i.e. the one in `/etc/`) against another (here we have used the EST timezone file against our `/etc/localtime -> /usr/share/zoneinfo/UTC`), showing us that EST is UTC-5 hours.
Next we will copy the suspect's `etc/timezone` to overwrite our own:

```
forensic@i:/RO-image# ls –l etc/localtime
-rw-r--r-- 1 root root 2944 Aug 23  2012 etc/localtime     # no symbolic link clue
forensic@i:/RO-image# cp etc/timezone /etc/timezone        # import the suspect's timezone file
forensic@i:/RO-image# date
Sat Nov 02 09:44:38 CET 2013
```

So again we have a suggestion that the suspect is on Central European Time. However, at the time and location of seizure CET (UTC+1) does not exist; it should be CEST (UTC+2). So we must set our clock to UTC+2.

Environment reconnaissance: clock calibration

We need to respect the timezone and keep our clock synchronized in order to conduct the investigation accurately. "Managing Accurate Date and Time", http://www.tldp.org/HOWTO/TimePrecision-HOWTO/tz.html

Now we have a choice – whether to calibrate our clock to (and hence show our results in) UTC, or whether we take the leap to present our results in CEST (saving the jury the effort of having to perform the +2 calculation). For the ease of presenting the walkthrough, we have chosen to calibrate to CEST, which later evidence may corroborate.

We choose to use the host operating system's timezone configuration utility:

```
forensic@i:/RO-image# dpkg-reconfigure tzdata
# we DON'T select Europe/Brussels (because at this time of year it runs on CET, not CEST)
# We know that CEST is UTC+2 hours (see:
# http://www.timeanddate.com/library/abbreviations/timezones/eu/cest.html)
# We must choose Etc -> GMT-2 (for lack of a UTC-2)
# N.B. that we have to choose -2 instead of +2 - it's a peculiar documented issue:
# "Bug#540305: tzdata: Etc/GMT+offset form is arcane, confusing",
# http://lists.debian.org/debian-glibc/2009/08/msg00036.html
# but checking the result, it is precisely what we want:
Current default time zone: 'Etc/GMT-2'
Local time is now:      Sat Nov 02 10:57:42 GMT-2 2013.
Universal Time is now:  Sat Nov 02 08:57:42 UTC 2013.
```

..so our local timezone is now set to CEST (UTC+2) all year round!
From here we can easily consider any alternate timezones (CEST=CET+1=UTC+2).
N.B. We need to ensure our clock is synchronized to ntp.pool.org throughout the investigation.

Environment reconnaissance: users

Checking `/etc/passwd` for additional users:

```
forensic@i:/RO-image# less /etc/passwd
```

We see there are plenty of users and the order in which they were created.
However, most of these user are system/application daemons and have no human operator.

We see that 'root' (home folder:`/root/`) was the first user created (probably by a human); the only other non-application related user appears to be 'alex' (home folder:`/home/alex`).

```
forensic@i:/RO-image# ls -F home
alex/
# -F flag adds a trailing '/' to signify directories and @ for symbolic links
```

We see that `/etc/passwd` lists only 5 `/home/` directories (home directories are usually associated with human users). Those are for `'root', 'alex' 'syslog', 'usbmux'` and 'saned'. Internet searching identifies all but the first 2 as system daemons. There are no other suspicious home directories listed.

It is unlikely that other human users existed on this machine (and were subsequently deleted), unless the attacker disguised their user as a common daemon or re-edited `/etc/passwd`. We use both `blkls` and `autopsy` later to search for any deleted directories in `/home`.

It's worth noting that the `/etc/passwd` implies the user installed applications for `'mail', 'printer spooling', 'news spooling', 'backup', 'irc' 'SANE scanner'`, and it seems the remaining line after the `'alex'` user is for a `'postgres'` administrator, so she may have installed it. This gives us clues into which applications we should research further to find trace evidence (e.g. printer spools, scans, proxy servers, IRC chats, postgres databases).

For example, the home directory of the backup daemon (`backup:x:34:34:backup:/var/backups:/bin/sh`) gives us a lead - we can compare our current `passwd` file with the only backup:

```
forensic@i:/RO-image# ls -l --full-time var/backups/passwd.bak etc/passwd && diff !#:3 !#:4
# "!#:n" cuts the nth word from the previous command (where ls is word 0)
-rw-r--r-- 1 root root 1746 2012-08-23 02:13:06.000000000 +0200 etc/passwd
-rw------- 1 root root 1669 2012-08-23 01:26:03.000000000 +0200 var/backups/passwd.bak
34a35
> postgres:x:115:125:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
```

The user `postgres` was added between `Aug 23 2012 01:26:03 & Aug 23 2012 02:13.06 EST.`

Researching may show us those elevated applications that do not store their profiles in the user's home directory (e.g. a shared printer might use a different logging place in `/usr/` or `/var/log` - the latter shows `dmesg` which illustrates the OS thought the laptop supported 'Bluetooth'). We'll come back to the `/var/dmesg` and other log files later.

```
forensic@i:/RO-image# ls -l --full-time var/backups/group.bak etc/group && diff !#:3 !#:4
# we care about when it was modified, not accessed
-rw-r--r-- 1 root root 886 2012-08-23 02:13:06.000000000 +0200 etc/group
-rw------- 1 root root 862 2012-08-23 01:26:03.000000000 +0200 var/backups/group.bak
49c49
< ssl-cert:x:110:
---
> ssl-cert:x:110:postgres
64a65
> postgres:x:125:
```

The user `postgres` obtained membership to the `ssl-cert` group between `Aug 23 2012 01:26:03 & Aug 23 2012 02:13.06 EST.`

`gshadow.bak` reveals more (we investigate the original `etc/gshadow` instead which we had overlooked):

```
forensic@i:/RO-image# less etc/gshadow
```

`alex` was a member of `sudo (superuser), dip` ("stands for *"Dialup IP". Being in group dip allows you to use a tool such as ppp..to dial up a connection"*, source: Debian default system groups description[2]), `plugdev` (use of external devices), `lpadmin` (printer administration) and `sambashare` (was she logging into a MS Windows share also? If so they need confiscation - or perhaps `alex` attacked them).

We also note that there is a backup copy of `/etc/gshadow` before the last change: `etc/gshadow-`.
This prompts us to wonder if we should search for other files appended with a dash, or even search for files with editor specific suffixes (e.g. `find /RO-image "*\~$"` would find all files ending in ~, which are `vi` editor temporary files. If we find out his editor preference, we could trace the files she edited but the editor has left backup or swap copies).

Environment reconnaissance: Checking /var

Add this later – needs severe trimming. Add /var/[233]

Applications of interest: email client

We find a local Mozilla Thunderbird email client configuration directory (`home/alex/.thunderbird`). We check profiles and deduce what application version(s) our suspect is using:

```
forensic@i:/RO-image# cd home/alex/.thunderbird
forensic@i:/RO-image/home/alex/.thundebird# less profiles.ini
# only one, "default", path: h9jaeoos.default
forensic@i:/RO-image/home/alex/.thundebird# sudo su  # enter administrator shell
root@i:/RO-image/home/alex/.thundebird# /RO-image/usr/bin/thunderbird -v
bash: /RO-image/usr/bin/thunderbird: Permission denied# querying some apps needs writable mount
root@i:/RO-image/home/alex/.thundebird# mkdir /NRO-image # create mount point – rarely used
root@i$PWD: unzip /home/forensic/case-FOR/derye.sda1.dd.zip –d /tmp/ # fresh copy
root@i$PWD: mount /tmp/derye.sda1.dd /mnt/NO-image  # mounted this image non-read-only
root@i:/RO-image/home/alex/.thundebird# /NRO-image/usr/bin/thunderbird –v
Thunderbird 14.0    # the command ran on a writable mount. Only used where necessary!
```

As an alternative to loading the entire virtual machine and 'alex' account into a *Live View*[3] session (which we later attempted), we instead import the profile directory into a working copy of Mozilla Thunderbird on the host. It's best to use the right version:

```
root@i:/RO-image/home/alex/.thundebird# apt-cache madison thunderbird # only v.16-17 available
```

So I downloaded a complete past package of the `thunderbird 14.0` from Mozilla & extracted it with the GUI. (*http://ftp.mozilla.org/pub/mozilla.org/thunderbird/releases/14.0/linux-i686/en-US/thunderbird-14.0.tar.bz2*)

```
root@i:/RO-image/home/alex/.thundebird# cp ../.thunderbird ~/
# make a working copy of alex's thunderbird profile my host user account home directory
```

Executed it, and found that my copied ~/.thunderbird was automatically imported.

Thunderbird 14 prompts for the Test Pilot addon - using the right version may lead us to deprecated addons/logs which no longer feature in later versions.

It is prudent to mark the ~/.thunderbird folder as read-only if copied (`find ~/.thunderbird -type f -exec chmod 444 {} \;`), provided it does lead to execution errors.

Headers

Thunderbird reveals each message's header (Other Actions -> View Source). We can review this source code, or utilize a parser tools (Email Header Analyzer, RFC822 Parser - MxToolbox, http://mxtoolbox.com/EmailHeaders.aspx) to make this easier to read.

---

[2] Debian default system groups description, http://wiki.gacq.com/index.php/Debian_default_system_groups_description

[3] *"Live View is a ..forensics tool that creates a VMware virtual machine out of a raw (dd-style) disk image or physical disk."* http://liveview.sourceforge.net. It didn't really work – so we created a dummy OS snapshot, and imported the entire FS into the dummy copy via the Gparted live CD (because the machine we were importing into could not be mounted during the import!)

When we analyze the header extract, the time is displayed as 8/22/2012 12:04:41AM in UTC; UTC+2 hours correlates to the 2:04AM CEST in our (Electronic only Thunderbird screenshot, Appendix D) **correctly**!

Headers: geolocation

message 2/4:
From: Elizabeth Queen Of Bohemia <elizabeth@oxfordian.info>
X-Originating-IP: 96.234.173.244 [<- verizon FiOS, Columbia, Maryland]
Date Tue, 21 Aug 2012 20:04:41 -0400
Received-SPF: neutral (google.com: 209.85.212.193 is neither permitted nor denied by best guess record for domain of elizabeth@oxfordian.info) client-ip=209.85.212.193 [<- Google server, CA]

message 3/4:
Received: from [172.16.77.129][<- internal network?] ([213.179.209.92][<- Amsterdam, Netherlands, ISP:SolidHost) by mx.google.com
User-agent: Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20120714 Thunderbird/14.0
From: Alexandrine de Rye <alexandrine@oxfordian.info>
Date Sat, 25 Aug 2012 21:02:56 +0200
Received from: 172.16.77.129 by mx.google.com

message 4/4:
X-Originating-IP: 96.234.173.244
From: Elizabeth Queen Of Bohemia <elizabeth@oxfordian.info>

We see that some headers are more interesting than others. Some attributes will geolocate the sender's location, whereas others will just refer to the mailserver or ISP used (from which we can seize additional data).

*"The **X-Originating-IP** is a de facto standard for identifying the originating IP address of a client connecting to a mail server.", X-Originating-IP, Wikpedia.org. http://en.wikipedia.org/wiki/X-Originating-IP*

Scouring databases

We should interrogate the profile's databases. Loading `Firefox's SQLite Manager` addon on the directory `.thunderbird/h9jaeoos.default`, we are especially interested in `signons.sqlite`,`places.sqlite` and `keydb.db` (Table *Files and folders in the profile - Thunderbird http://kb.mozillazine.org/Files_and_folders_in_the_profile_-_Thunderbird*).

```
places.sqlite: SELECT * FROM moz_historyvisits
# no records. It appears this entire DB is a remnant from firefox, and doesn't need to be in thunderbird. That said, it may illustrate clicked on hyperlinks.
form_history.sqlite: SELECT * FROM moz_formhistory,moz_deleted_formhistory
cookies.sqlite: SELECT * FROM moz_cookies     # 1 record: google.com
```

`key3.db` is an encrypted password database, a we can recover passwords from the Thunderbird 14.0 GUI itself:
Menu: Edit -> Preferences -> Security -> Passwords -> Saved Passwords... -> Show Passwords

Gmail : to login or not to login?

and now we consider if we want to login to http://www.oxfordian.info's Gmail account (knowing Gmail stores and updates the last login date). We decide we will (with the proper warrant: see assumptions) so we can geolocate the history of recent login locations, and find any contacts and further evidence.

The welcome message gives us a clue of the portal's administrator: *This is a resource for the Software & Security Masters course Forensics module at the Computing Laboratory of Oxford University, created by Gareth Digby*", and we might retrieve more our suspect's network and office hardware by contacting them.

Alex hasn't logged in recently (the only visible record represents our login – captured in Appendix D). The password was "*Changed over 1 year ago on 21 August 2012*", either by Alex or the site administrator. There are no additional chat transcripts. We might guess that Alex probably hasn't logged in since June, since we are greeted to a tutorial which was rolled out in May (http://www.businessinsider.com/new-gmail-2013-5).

Confirming that the Gmail IMAP hierarchy matches that of Thunderbird, we proceed to check the account Chat history (empty) and any mail/forwarding filters that may intercept IMAP downloads, and then move onto additional Gmail menu items (their Google+ profile, their calendar, their drive/shared documents, their maps, their groups etc.) If we search the calendar for *, the events (all template public celebratory days provided by Google) we see they lead back to August 2012, which may be when the calendar was created.

Otherwise we see that Alex is not a member of any groups, has no contacts, has not photos or maps (*"You haven't saved or rated any places yet..You have no maps..Create a map of your favorite vacation spots"*). There might be other Google apps like Sites, Shopping or Finance with a history trail which could be explored.
Note that this is quite dangerous territory as the investigator now changing online activity logs (which could void any presented evidence), and in real scenarios would require appropriate permission before trespassing on the crime scene (ideally a working copy would be obtained from Google).

Applications of interest: internet browser

We find a local Mozilla Firefox internet browser configuration directory (`home/alex/.mozilla/firefox`). We check for additional profiles to load:

```
root@i:/RO-image/home/alex/.thundebird# cd /RO-image/home/alex/.mozilla/firefox
root@i:/RO-image/home/alex/.mozilla/firefox# less profiles.ini
# only one, "default", path: wbaoqjsm.default
root@i:/RO-image/home/alex/.mozilla/firefox# /NRO-image/usr/bin/thunderbird –v
Mozilla Firefox 22.0
# deduce what application version our suspect is using to guide our method
# we can run 'whereis firefox' on the host to list of associated directories containing trace
```

We wanted to load a read-only copy of the profile into our host version so as not to contaminate the results, so we set the default path of `/root/.mozilla/firefox/profiles.ini` to the read-only `/RO-image/home/alex/.mozilla/firefox/wbaoqjsm.default` and attempted to load a Firefox interactive session.

However, Firefox refused to load with a read-only profile folder. Setting `/NROimage/home/alex/.mozilla/firefox/wbaoqjsm.default` loaded fine however.

This wasn't professional for investigation, as our every page load affected the browsing history. Not wanting to apply date filters to our exported results, we instead read the database files with our `Firefox SQLite Manager` addon.

```
/RO-image/home/alex/.mozilla/firefox/wbaoqjsm.default/places.sqlite:
# we scoured every database & and every table, presenting the best columns in CEST.

SELECT
datetime(moz_historyvisits.visit_date/1000000,'unixepoch','localtime'),moz_places.url,moz_place
s.visit_count
FROM moz_places,moz_historyvisits
WHERE moz_places.id=moz_historyvisits.place_id
ORDER BY datetime(moz_historyvisits.visit_date/1000000,'unixepoch','localtime') ASC
```

"The `localtime` modifier assumes the time string to its left is in Universal Coordinated Time (UTC) and adjusts the time string so that it displays `localtime`"[4] - we have assumed this also, as this is the default behaviour of Firefox. *"A Mozilla application has no clock of its own. It uses your system clock. Your system clock is controlled by your operating system settings..To check your system time in a Mozilla application, from the main menu bar choose Tools*

[4] http://www.sqlite.org/lang_datefunc.html

– Web Development – JavaScript Console..paste Date(). The JavaScript Console displays your current date, local time, offset from GMT and time zone."[5] and time and date (browser and OS) is set reliably within milliseconds to CEST (confirmed online). An intelligent Alex might have modified her OS / browser timezone to obscure results.

The query was exported as a CSV document ([**Appendix B: Browser history**]). It shows the chronological order of webpage visits (and the visit count for each page). *Column headings & line numbers were added manually.*

When we correlate these times to those given by our ls command later we will expect to see cache files line up!

`moz_inputhistory` sounds interesting ("*moz_inputhistory - A history of URLS typed by the user*" [6])
Let's also export the bookmarks:

```
SELECT title, datetime(dateAdded/1000000,'unixepoch', '-4
hours'),datetime(lastModified/1000000,'unixepoch', '-4 hours')
FROM moz_bookmarks,moz_bookmarks_roots        # both tables are the same
```

There is nothing outside of the standard distribution here.
`cookies.sql` could have been quite interesting (but wasn't), but `signons.sqlite` is really exciting: two chosen images in Appendix D illustrate information that we could have never retrieved from interactive GUI browsing alone.

This is our most insightful find. See the following fields:

```
SELECT hostname, formsubmitURL,userNameField, passwordField,
datetime(timeCreated/1000000,'unixepoch', '-4
hours'),datetime(timeLastUsed/1000000,'unixepoch', '-4
hours'),datetime(timePasswordChanged/1000000,'unixepoch', '-4 hours'),timesUsed
FROM moz_logins
```

output:

```
hostname                    URL of submitted form    user   p/w  time created
"http://pepys.dyndns.org","http://pepys.dyndns.org","log","pwd","1970-01-16 09:51:54",
time last used          time p/w changed    times used
"1970-01-16 09:52:02","1970-01-16 09:51:54","2"
```

We also note the date 1970-01-16, perhaps the server (or username) had been created 16 days previously?

`formhistory.sqlite` and `healthreport.sqlite` offer no admissible evidence.
`downloads.sqlite` is empty (although we must also check any filesystem download folders in case the entry was deleted from the browser log).

we also investigate the following files in case bookmarks were deleted:

```
root@i:/RO-image/home/alex/.mozilla/firefox# diff -s wbaoqjsm.default/bookmarkbackups/*
Files wbaoqjsm.default/bookmarkbackups/bookmarks-2012-08-23.json and
wbaoqjsm.default/bookmarkbackups/bookmarks-2012-08-25.json are identical
root@i:/RO-image/home/alex/.mozilla/firefox# cat wbaoqjsm.default/bookmarkbackups/bookmarks-
2012-08-25.json     # standard; although Ubuntu One might suggest an online storage account.
```

Instead of SQLlite Manager CSV or txt exports we might have used a Firefox addon like the deprecated `history-export` to print the history timeline for court (were the provided addon trusted by industry or formally rigid), but we would have missed out fields like `inputhistory` and `formsubmitURL` had we not interrogated the SQL database in more depth.

Steganography
We notice that the URL `http://pepys.dyndns.org/wp-content/uploads/2012/08/200px-Prinsenvlag.svg.png` was uploaded to the victim's server. We submit `200px-Prinsenvlag.svg.png` to a familiar

[5] http://kb.mozillazine.org/Time_and_time_zone_settings
[6] *http://kb.mozillazine.org/Places.sqlite. Entity relationships: http://people.mozilla.org/~dietrich/places-erd.png*

decoding website[7]: Result: a pattern of db repeating, ending in df (precise regexp: (db)*df; loose regexp: [db]*f). It reminds us of the repeating patterns found when one conceals an image in HTML, but since this is not HTML output (utilizing tags and RGB values) we believe there is no concealed CSS image.

```
forensic@i:/RO-image/home/.mozilla/firefox# cd ../../; file Downloads/200px-Prinsenvlag.svg.png
Downloads/200px-Prinsenvlag.svg.png: PNG image data, 200 x 133, 8-bit/color RGBA, non-
interlaced
```

It certainly is a .png image (even after renaming). The original svg file may have contained a hidden message (e.g. as an XML comment, "<!-- like this -->").

```
forensic@i:/RO-image/home/# less Downloads/200px-Prinsenvlag.svg.png # also reveals there is no
svg code (http://www.w3schools.com/svg/svg_example.asp).
```

Researching the double extension it appears that the image was created a command similar to:
`qlmanage -t -s 200 -o . 200px-Prinsenvlag.svg` ('*quicklook*' is also used by the macosx preview utility)
....but the reversal from this png image back to the original svg file does seem to be possible.

Applications of interest: instant messenger

`usr/lib/purple-2/, usr/share/purple/` suggests that Instant Messaging software (such as 'Pidgin') as part of the OS distribution. While there is not local user configuration directory, there may be certificates or chat logs of interest in the `usr/share/purple/` folder. Let's compare it to our baseline (host) directory:

```
root@i:/RO-image/home/alex/.mozilla/firefox# cd /RO-image/usr/share/purple
root@i:/RO-image/usr/share/purple# find ca-certs/ -type f -exec diff -s ./{}
/usr/share/purple/{} ";" # compare files in 'ca-certs' (suspect vs. host)
Files ./ca-certs/AOL_Member_CA.pem and /usr/share/purple/ca-certs/AOL_Member_CA.pem are
identical
Files ./ca-certs/DigiCertHighAssuranceCA-3.pem and /usr/share/purple/ca-
certs/DigiCertHighAssuranceCA-3.pem are identical
..
```

.. all 8 .pem files are identical. They are verified online as standard distribution public key examples.

Applications of interest: SSH

We find `.ssh/known_hosts`, which lists the keys for 2 servers to which the suspect has previously connected.
We want to match these keys with `pepys.dyndns.org`, `strongvpn.nl` or some other machine, but the hostnames are hashed (each entry starts with '`|1|`').

We can verify our guesses:

```
root@i:/RO-image/usr/share/purple# cd /RO-image/home/alex/
root@i:/RO-image/home/alex/# ssh-keygen -H -F pepys.dyndns.org -f .ssh/known_hosts
# Host pepys.dyndns.org found: line 1 type RSA
|1|8iMPQjDuaVt ...
```

But the other URLs are incorrect. We can use a tool to bruteforce it, but before we do let's try the IP address we found in `var/` (remember line "`shellAauxiliary/scanner/ssh/ssh_loginSSSH samuel:elizabeth (96.234.173.244:22)`" from when we were *grepping* for IP addresses):

```
root@i:/RO-image/home/alex/# ssh-keygen -H -F 96.234.173.244 -f .ssh/known_hosts
# Host 96.234.173.244 found: line 2 type RSA
|1|HSSnUj2bBKE...
```

Correct! 'alex' connected to both hosts via SSH and saved them. We must check with Mr. Pepys whether they match the target, or perhaps a third 'reinforcements' server from which the attacker downloaded an attack toolkit.

[7] http://utilitymill.com/utility/Steganography_Decode

Applications of interest: keyrings

We notice `.gnome2/{keyrings/login.keyring, user.keystore}`, and the applications `gnome-keyring` & `gnome-keyring-3` seem related. A quick internet search reveals a GUI front end call Seahorse: *https://help.gnome.org/users/seahorse/stable/seahorse-getting-started.html.en*

Process (all tasks run as root to avoid permissions issues):
Import `gnome2` configuration; run `seahorse`; if locked, crack password; retry `seahorse`.

```
root@i:/RO-image/home/alex/# cp -r /RO-image/home/alex/.gnome2 ~/
root@i:/RO-image/home/alex/# cp -r /RO-image/home/alex/.gnome2/keyrings/login.keyring
~/.local/share/keyrings/  # the newest version of seahorse defaults to this location
root@i:/RO-image/home/alex/# seahorse # run GUI keyring manager
```

It's locked with a password, and none of our keyword guesses have worked. We really want to retrieve any saved keys of third party servers!

Unfortunately for alex, there are tools that can bruteforce `gnome-keyrings` with a wordlist[8]:

```
..
root@i:/RO-image/home/alex/# cd ~/JohnTheRipper-unstable-jumbo/run
root@i:/root/JohnTheRipper-unstable-jumbo/run # cp -r /RO-image/home/alex/
.gnome2/keyrings/login.keyring ./
root@i:/root/JohnTheRipper-unstable-jumbo/run# less password.list # check supplied wordlist
root@i:/root/JohnTheRipper-unstable-jumbo/run# ./keyring2john login.keyring > hash # convert
root@i:/root/JohnTheRipper-unstable-jumbo/run# time ./john ./hash # now process converted hash
Loaded 1 password hash (GNOME Keyring iterated-SHA256 AES [32/32 OpenSSL])
black            (login.keyring)
guesses: 1  time: 0:00:00:18 DONE (Sun Nov  3 11:43:17 2013)  c/s: 441  trying: black
Use the "--show" option to display all of the cracked passwords reliably

real   0m18.727s
user   0m16.793s
sys    0m0.936s
root@i:/root/JohnTheRipper-unstable-jumbo/run# ./john --show ./hash
login.keyring:black
1 password hash cracked, 0 left
```

Let's attempt alex's password:

```
/root/JohnTheRipper-unstable-jumbo/run# cp /RO-image/home/alex/etc/{passwd,shadow} .
# both /etc/ passwd & shadow files required
/root/JohnTheRipper-unstable-jumbo/run# ./unshadow passwd shadow > passwd_hash
# unshadow the passwd file
/root/JohnTheRipper-unstable-jumbo/run# time ./john ./passwd_hash # now process converted hash
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 1 password hash (sha512crypt [32/32 OpenSSL])
black            (alex)
guesses: 1  time: 0:00:03:02 DONE (Sun Nov  3 11:51:31 2013)  c/s: 129  trying: black
Use the "--show" option to display all of the cracked passwords reliably

real   3m3.278s
user   2m47.698s # it takes a bit longer for etc/passwd
sys    0m7.140s
```

Now we can enter that *Live View*. We also unlock alex's `gnome-keyring` manager with `seahorse` but find it empty.

[8] https://github.com/magnumripper/JohnTheRipper/blob/bleeding-jumbo/doc/README.keyring

Filesystem listings

Now we come to the most important part of our investigation. We want to illustrate the contents of our filesystem (selectively and relevantly), using the appropriate command for the job.

The two command at our disposal are `ls` and `find` (GNU versions).

When one uses the `ls` command (organizing by either file or modification times), the output is grouped by directory. That can be visually appeasing; but when listing subdirectories, it struggles to provide a concise chronological timeline of events (due to the sorting being performed subdirectory by subdirectory).

ID	Command	Sorted by	On	Why
1	ls	modification	/	'/' is vast; files from different applications are mixed together. sorting files by subdirectories makes for easy targeted lookup of a particular application (e.g. /opt/metasploit-framework/)
2	ls	access	/	
3	find	modification	/root/	A home directory is manageable; activity by a single user is grouped together. find provides a concise chronological timeline of user activity (personal file modification and read access: *class A*). In a home directory it also provides a concise chronological timeline of application activity (application download and modification times, and application access (or run) times*:class B*)
4	find	access	/root/	
5	find	modification	/home/alex/	
6	find	access	/home/alex/	

N.B. Exports list user forensic which has equivalent UID of 1000 as user alex on the suspect's machine

Our output files take the following syntax: **<command>_<directory>_<sort type (access or retrieval)>.txt**

```
cd /RO-image

1       Entire filesystem modification times
        ls -aRltF --full-time ./  > ~/ls_all_modification_times.txt
        sort by modification times (seperate sub-directories)
        show complete timestamps for previous years

        URL: http://users.ox.ac.uk/~kell3138/FOR/all_modification_times.txt, 35MB

2       Entire filesystem access times
        ls -aRltuF --full-time ./  > ~/ls_all_access_times.txt
        sort by access times (segregate sub-directories)
        show complete timestamps for previous years

        URL: http://users.ox.ac.uk/~kell3138/FOR/all_access_times.txt, 35 MB

3       root home directory modification times
        find root -type f | xargs ls -lt  --full-time > ~/find_root_modification_times.txt
        Create a user timeline (sorted by modification times)

        URL: http://users.ox.ac.uk/~kell3138/FOR/find_root_modification_times.txt, 238b


4       Root home directory access times
        find root -type f | xargs ls -ltu  --full-time > ~/find_root_access_times.txt
        Create a user timeline (sorted by access times)

        URL: http://users.ox.ac.uk/~kell3138/FOR/find_root_access_times.txt, 238b
```

```
5      alex home directory modification times
       find home/alex -type f | xargs ls -lt  --full-time > ~/find_alex_modification_times.txt


6      alex home directory access times
       find home/alex -type f | xargs ls -ltu  --full-time > ~/find_alex_access_times.txt
```

With these 6 exports to hand we can form our opinions of the events that took place more quickly.

Commands 1-4 run beautifully.
HOWEVER, we need to modify commands 5 & 6 just slightly! Their principle remains the same.
This is because commands 5 & 6 report two types of warning:

```
ls: cannot access ./.thunderbird/Crash: No such file or directory # xargs can't handle spaces[9]
unable to execute /bin/ls: Argument list too long                 # the input list is too long[10]
```

We research our two errors, and are advised the following should work (replacing command 6):

```
find alex -type f -print0 | xargs -0 ls -ltuF --full-time > ~/find_alex_access_times-
RESPAWNERROR.txt    // no error messages reported.
```

It appears to be sorted correctly, but wait, it's not:

```
-r--r--r-- 1 forensic forensic     84 2012-08-23 02:50:02.000000000 +0200
home/alex/Development/nmap/libpcap/ChmodBPF/.svn/text-base/StartupParameters.plist.svn-base
-rw-rw-r-- 1 forensic forensic    6532 2012-08-25 20:54:24.000000000 +0200 home/alex/hacks/mail-
thread.txt
-rw-rw-r-- 1 forensic forensic    3506 2012-08-25 20:41:27.000000000 +0200 home/alex/hacks/mail-
thread.txt~
```

N.B. the owner and group are set to forensic (not present on 'hacker-box') as the UID (1000) of the 'alex' account matches 'forensic' on our host system.

For the most part the sorting works, but for some reason it has a transition in the middle where the sorting 'resets' (this just occurs once because `ls` respawns in the middle). This will not do.
This is because the memory buffer is not large enough to handle sorting all of these files at once[11], so the sorting action resets. Therefore when `ls` was previously complaining that it could not handle the input, a 'hack' workaround was not the solution. It's a shame because the output was concise and to the point.

Our options are to:
increase the kernel's `ARG_MAX` (to increase the number of command line arguments `'xargs ls'` can take)[3];
use find's "printf %" options to print exactly what we want instead of using ls.
pipe our find output to another sorting program, because piping does not pass arguments (it is unaffected by the POSIX `ARG_MAX`[12]). This allows us to sort our long `ls` output without resorting to `xargs`.

So finally, the proven replacement commands are:

```
5      alex home directory modification times
       find home/alex -type f -exec ls -lt --full-time {} \; | sort -t' ' -k +6,6 -k +7,7 >
       ~/find_alex_modification_times.txt
```

---

[9] *"with the version below find will not fail: "find /path -type f -print0 | xargs -0 ",* http://en.wikipedia.org/wiki/Xargs
[10] Beyond Arguments and Limitations, http://www.linuxjournal.com/article/6060
[11] GNU Core Utilities FAQ, http://www.gnu.org/software/coreutils/faq/coreutils-faq.html#Argument-list-too-long
[12] *"This limits is a safety for both binary programs and your Kernel...There is no such limit on pipe size."* Stack Exchange Forum, http://unix.stackexchange.com/questions/38955/argument-list-too-long-for-ls

URL: http://users.ox.ac.uk/~kell3138/FOR/find_alex_modification_times.txt, 677Kb

6 alex home directory access times

```
find home/alex -type f -exec ls -ltu --full-time {} \; | sort -t' ' -k +6,6 -k +7,7 > ~/find_alex_access_times.txt
```

URL: http://users.ox.ac.uk/~kell3138/FOR/find_alex_access_times.txt, 677Kb

With many thanks[13] we did not have to resort to sorting with a GUI spreadsheet.

Now let's clean it up for readability (awk, cut, tr, sed):

```
root@i:/RO-image# sed -i.bak s/\.000000000\ \+0200//g find_alex_modification_times.txt
# removes repeat strings for monitor readability.
awk '{$1=$2=$3=$4=$5=""; print}' find_alex_modification_times.txt  | sed 's/^ *//g' | nl > find_alex_modification_times_printable.txt
For printability, cut first 5 columns (until timestamp) & trim whitespace, and number lines.
```

URL: http://users.ox.ac.uk/~kell3138/FOR/find_alex_modification_times_printable.txt, 456Kb

A trimmed, colour coded and commented version is our timeline in Appendix C :)

Shell logs

```
root@i:/RO-image# less home/alex/.msf4/logs/framework.log  # history of metasploit commands
root@i:/RO-image# less home/alex/.bash_history             # history of bash terminal commands
```

Zeitgeist and gedit (GTK+)

Let's reload alex's Zeitgeist profile in order to deduce which files were opened via the GUI (i.e. by icon clicking).

Zeitgeist: "*Zeitgeist is a service which logs the user's activities and events.*"
"*How can I view the activities?*"
"*By using the GNOME Activity Journal.*" http://wiki.zeitgeist-project.com/Frequently_Asked_Questions

```
root@i:/RO-image# cp -r home/alex/.local/share/zeitgeist/ ~/.local/share/zeitgeist # import
root@i:/RO-image# zeitgeist-daemon & # start zeitgeist
root@i:/RO-image# gnome-activity-journal # view imported profile

Illustrated in main report.
```

Let's open home/alex/.local/share/zeitgeist/activity.sqlite too:

```
checked every table, found 2 interesting:
SELECT uri.value,text.value from uri,text     # export to activity.csv

root@i:/RO-image# uniq activity.csv    # display unique lines
```

We get a good picture of what 'alex' was clicking on via GUI. The table 'event' also includes a timestamp, but dates start from 1970, so it only really tells us the interval.

```
root@i:/RO-image# strings home/alex/.local/share/zeitgeist/*\.DB
```

..also showed the terms above (but not so usefully).

[13] How to list files sorted by modification date recursively, Stack Overflow, http://unix.stackexchange.com/questions/9247/how-to-list-files-sorted-by-modification-date-recursively-no-stat-command-avail. This took a lot of research.

gedit (GTK+ application)

Let's examine the GTK+ applications shared recently used list:

```
root@i:/RO-image# .local/share/recently-used.xbel
home/alex/hacks/pepys.dyndns.org.reconn.txt" added="2012-08-25T16:11:03Z" modified="2012-08-25T17:40:34Z"
visited="2012-08-25T16:11:04
home/alex/hacks/mail-thread.txt" added="2012-08-25T18:33:31Z" modified="2012-08-25T19:02:44Z"
visited="2012-08-25T18:33:32Z
home/alex/hacks/john/mbox" added="2012-08-25T18:33:05Z" modified="2012-08-25T18:33:05Z" visited="2012-08-
25T18:33:05Z
home/alex/hacks/samuel/samuel" added="2012-08-25T18:40:03Z" modified="2012-08-25T18:40:03Z"
visited="2012-08-25T18:40:03Z
home/alex/Downloads/200px-Prinsenvlag.svg.png" added="2012-08-25T19:17:17Z" modified="2012-08-25T19:28:04
visited="2012-08-25T19:17:17Z"
```

Let's compare our filesystem modification dates against the "`modified=`" fields to find which files were modified in `gedit` (and no other subsequent program):

```
root@i:/RO-image# cd home/alex/hacks
root@i:/RO-image# ls -l --full-time  pepys.dyndns.org.reconn.txt mail-thread.txt john/mbox
samuel/samuel ../Downloads/200px-Prinsenvlag.svg.png | awk ' { print $6,$7,$9 } '
2012-08-25 21:17:18.000000000 ../Downloads/200px-Prinsenvlag.svg.png
2012-08-25 20:28:45.000000000 john/mbox
2012-08-25 20:48:15.000000000 mail-thread.txt
2012-08-25 19:40:34.000000000 pepys.dyndns.org.reconn.txt
2012-08-25 20:30:03.000000000 samuel/samuel
```

As an example, comparing the output to `'recently-used.xbel'` (after applying an offset of -2 to our output so it becomes UTC) shows:

recently-used.xbel Log	**vs. ls output (converted to UTC)**
`home/alex/hacks/pepys.dyndns.org.reconn.txt "modified="2012-08-25 17:40:34"`	`2012-08-25 17:40:34` **`# match!`**
`home/alex/hacks/mail-thread.txt modified="2012-08-25 19:02:44"`	`2012-08-25 18:48:15`
`home/alex/hacks/samuel/samuel modified="2012-08-25 18:40:03"`	`2012-08-25 18:30:03` **`# 10 minutes out`**
`home/alex/hacks/john/mbox modified=2012-08-25 18:33:05"`	`012-08-25 18:28:45` **`# not quite`**
`home/alex/Downloads/200px-Prinsenvlag.svg.png added="2012-08-25 19:17:17"`	`2012-08-25 19:17:18` **`# 1 second out`**

So we might interpret this as telling us that "`pepys.dyndns.org.reconn.txt`" was modified by a GTK+ program and not subsequently edited.

So as not to go over the Investigation report page limit, I have included some bonus research in [Appendix E: Extra] for interest because it was a shame to waste the documentation.