Exploring a Solution for Retrieving Stolen Devices Using Digital Means
Jelle Jan Bankert
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.j.m.bankert@student.utwente.nl

ABSTRACT
In this paper a solution for the Dutch police to retrieve stolen electronic devices, in particular laptops, will be explored. The proposed solution is determining the ip-address of a stolen device and then using 126na Sv. 126na Sv allows the police to match the ip-address to a physical address. Two practical methods for retrieving the ip-address will be discussed and the overall viability of the proposed solution is also tested. This is done by planting specially prepared laptops in environments known to handle stolen devices and then monitoring them. From the results it is concluded that the proposed solution is viable.

Keywords
stolen device retrieval, digital policing, ip-address retrieval, 126na Sv, CIOT Information System

1. INTRODUCTION
The Enschede police department aims to create a new ‘playground’ in which digital volunteers can operate, try out approaches to digital policing and see which ones are effective. The most effective approaches can then be implemented in a larger, possibly national, context. This playground is used to explore a new solution for solving burglary, one of the main points of focus for the Dutch police.

A recent news report shows that the number of burglaries has shown a large increase in the last year and only 12% of cases were solved [2]. Break-ins can have severe psychological consequences, besides the obvious financial ones. The Enschede police have stated that 80% of all burglaries are performed by 20% of burglars. This would mean that solving a limited number of cases and putting these criminals away, could lead to a big drop in break-ins. The police also believe that most of these burglars are looking for a quick way to get money, and are not necessarily thinking through every step, during or after their crime. Since they are currently not being traced digitally, this means that even ‘smart’ criminals probably overlook ways in which they can be digitally found by a computer expert.

This has led us to focus on solving burglary using digital means, in order to raise the number of solved cases and decrease the number of burglaries.

In order to make this paper accessible to computer science students and professionals as well as policemen, jargon and technical terms are explained in the glossary, Chapter 8.

1.1 Problem Statement
In a talk at the international computer security conference Defcon 18, a man known by his alias ‘Zoz’ showed how a computer that was stolen in a break-in, was retrieved two years later [16]. This was possible because the computer had been connected to the internet without any modifications and it happened to run DynDNS, which revealed its ip-address. It was then possible for him to derive the computer’s physical location and retrieve the computer. More generally, ‘Zoz’ used the strategy of determining the ip-address of a stolen device and then matching that to a physical address in order to retrieve it.

This solution is successful when three conditions are satisfied: a computer connects to the internet after being stolen, this computer has software that reveals its ip-address through the internet and the ip-address can be matched to a physical location.

The problem this paper focuses on, is solving burglary using digital means. Rather than defining a new solution to solve this problem, the viability of the solution proposed by ‘Zoz’ will be evaluated.

In the case of ‘Zoz’ the computer’s location could be determined because the new user had saved his own address on the computer. ‘Zoz’ could access this data through the SSH and VNC services he had installed before the computer was stolen and which were now available at the retrieved ip-address. However, the police do not have to rely on preinstalled software and saved physical address for matching the ip-address to a physical address. The police have access to the ‘Centraal Informatiepunt Onderzoek Telecommunicatie’ CIOT by way of the CIOT Information System (CIS). This provides a 91% hit rate [11].

In this paper the viability of the proposed solution for digital policing is explored. The paper will focus on retrieving laptops, because this is the type of device available for experimentation.

The contribution of the proposed solution will be increasing the number of solved burglary cases. Currently 12% of burglaries are solved and therefore even increases in the order of magnitude 1% will still be significant.

1.2 Research Questions
This leads to the main research question: what is the viability of retrieving stolen laptops by retrieving their ip-address and matching that to physical address?

The main question is divided into three parts
1. What is the chance of a stolen computer being connected to the internet?
2. What commonly used software can be used to retrieve the ip-address?
3. How is an ip-address matched to a physical address?

These three questions will be answered in three parts. Chapter 2 treats the methods and result of the experiment to answer question 1 and this is the main focus of this paper. Three laptops are prepared with specially crafted software and simulate being distributed as stolen devices. Chapter 3 will treat retrieving ip-addresses. It will in particular focus on the concrete cases of using the Gmail webpage and Skype client. A small experiment is performed to evaluate the merits of monitoring Skype’s network connections and the Skype log for retrieving ip-addresses. Chapter 4 will treat the legal framework for matching an ip-address to a physical address. In particular 126na of the Dutch penal code ‘Wetboek van Strafvordering’ (126na Sv).

1.3 Related Work
The first steps in digital policing in the Netherlands were taken in 2006. In that year Team High Tech Crime (THTC) was established. THTC currently consists of 60 persons and they focus on a limited number of high-profile, high-tech cases. Between 2006 and 2011 32 cases were investigated [3]. Currently digital policing pilots exist in many regions in the Netherlands. Relevant to this paper are projects by the Groningen police and police district Gelderland-Zuid. They are both setting up projects with lure laptops outfitted with a gps tracer [7], [13]. In contrast to the pilots in Groningen and Gelderland-Zuid, this paper focuses on retrieving laptops through their internet connection rather than with a GPS tracer. This will in theory allow the retrieval of any laptop, rather than only specific lure laptops.

2. EXPERIMENT
The main focus of this paper is experimentally finding the chance that a stolen laptop reconnects to the internet. The experiment’s method consists roughly of three stages. In stage one the laptops are prepared and outfitted with tracking software, sections 2.1 and 2.2. Stage two is distributing the laptops into (simulated) criminal environments, sections 2.3 and 2.4. In the final stage a tracking server will log the devices that connect to the internet, section 2.5.

2.1 Devices
In order to obtain devices for the experiment the Distributed and Embedded Security group (DIES), the Pervasive Systems group (PS), the ICT Service center (ICTS) and the Notebook Service Center (NSC) were all contacted. ICTS responded via DIES and has kindly provided three laptops. They were the only group to provide laptops and this poses a problem for statistical analysis. So, rather than quantifying the results, they will be qualitative in nature, an indication of the viability of the proposed solution.

The three provided laptops are all about 10 years old, or more intuitively, all three have a built in floppy disk drive. Two laptops are Asus type L3500Tp, introduced in 2002 and the last laptop is an Asus type L5800C, introduced in 2003. The first two were marked as ‘Marianne’ and ‘Yvonne’, for ease of distinction, and we called the L5800C ‘Anne’.

As possible choices for the operating system only Microsoft Windows is considered because the aim is to simulate the most common case. All three laptops are powerful enough to run Windows XP. ‘Yvonne’ has only 480MB of RAM so installing a newer, more attractive operating system is not possible [10]. For ease of maintenance and for the sake of keeping all three laptops as equal as possible, Windows XP is chosen as the operating system for this experiment.

The three laptops all have a dead battery, meaning that without a power cable the laptops will immediately power off. Additionally wireless connectivity support on the two L3500Tp laptops is poor. They lack support for WPA and WPA2 encrypted wireless networks, even with service pack 3 for Windows XP installed. Furthermore after the computer is booted, it is necessary to reboot the wireless driver before any wireless connection can be established.

It must be noted that these shortcomings are expected to influence the measurement, especially the poor wireless connectivity support in the L3500Tp’s. The shortcomings could pose a barrier to an inexperienced user for connecting the laptops to the internet.

2.2 Tracking Software
To track whether or not a laptop is online, a particular kind of software is necessary. Many companies offer software to track a laptop after it is stolen, for example LoJack [1], PC Phone Home [5] and Adeona [14]. There are however two very practical problems with the available solutions. The first is that most solutions cost money and there was no budget available for this paper. Adeona is a free, open source, tracking solution that uses OpenDHT to store data in distributed hashtables without the need for centralized servers. Adeona was developed at the University of Washington in collaboration with the University of California San Diego and the University of California Davis. In a 2009 announcement they stated that the OpenDHT system is no longer available and thus Adeona isn’t able to operate [9]. This means that no ready solutions exist and custom software is required to be built.

This new software is built in two parts, a client and a server. The client is placed on the laptops and the server software on a computer that we monitor. The programs are written in Java because we are experienced in writing programs with network functionality in this language.

2.2.1 The Client
The client has these requirements:
1. The client connects to the server as soon as the computer has an internet connection
2. The client manages a backlog of its network status when it can’t reach the server
3. The client identifies itself to the server when it is connected
4. The client sends its entire backlog to the server as soon as it connects to it
5. The client sends regular messages to the server while it is connected
6. The client is hard to detect by the user of the stolen laptop

In order to satisfy requirement 1, the client executes the command to connect to the server in a loop without pause. Furthermore the client is started as soon as the user logs in. This is a slight compromise from starting the client as soon as the computer is booted. Within the given time this was a necessary compromise and of small consequence. The laptops are configured to log in as soon as they are booted, so in practice there is little difference.

As long as the client isn’t connected to the server, every 30 seconds a status message is written to the backlog, to satisfy requirement 2. This interval of 30 seconds is built in to decrease the speed the log file grows at. With the current configuration this adds up to a log file of roughly 256 KB per 24 hours that
the laptop is turned on and not connected to the server. An example of the formatting of the status updates that the client saves to its backlog is shown here:

Could not connect to remote host, info: 192.168.2.11 (marianne), 2013/05/22 18:30:25

Notice that that the client saves its local ip and local time. The name in parentheses is the resolved host name.

Once the client connects to the server, it sends a message in which it identifies which of the three laptops it is running on, this satisfies requirement 3. This identifier is the current logged in user, which is set during preparation. After the client has identified itself, it immediately sends a message with the entire backlog, satisfying requirement 4. It is likely that connecting the laptop to the internet will be done within its first 24 hours of use. With the current broadband internet the entire backlog should be sent within a second and should be unnoticeable to the user. The backlog file is cleared after its contents have been sent to the server.

After the backlog has been sent to the server, the client sends its status every 5 seconds to satisfy requirement 5. This leads to a log file size increase of 2MB per 24 hours of regular updates. This poses is no problem for today’s hard drives and the used bandwidth is unnoticeable on any internet connection. The formatting of the regular updates during a connection is very similar to the status messages in the backlog:

192.168.2.11 (marianne), 2013/05/22 18:42:21

To satisfy requirement 6 several precautions have been taken. Firstly the startup folder, in which the shortcut is placed, has been set to hidden. The program runs, but is invisible as long as ‘hidden’ folders are set to be hidden in the Windows Explorer settings. Secondly the Java program is packaged as a .jar file. This ensures that no windows are opened to run the program. As a side effect the process is only described as javaw.exe in the Windows Task Manager, which helps to obfuscate the true nature of the program. The program uses about 10MB of RAM and 0% CPU time after rounding by the Task Manager. This allows the client to remain unnoticed even further. The actual program and log files are placed in a system folder. As a last line of defense the program is called UAC.jar and the log file winupdate.txt to give the impression that the files are default windows components.

2.2.2 The Server

The server has these requirements:

1. The server accepts incoming connections from the client
2. The server handles multiple clients simultaneously
3. The server logs data sent by the client
4. The server has a high uptime

In order to satisfy requirement 1, the server uses a serversocket that accepts incoming connections. Every accepted connection is then handled in newly created, separate thread. This ensures that the original accepting thread is free to accept more clients, satisfying requirement 2.

To satisfy requirement 3, the server writes all data to its disk. The identifier sent by the client is used as the filename and all data from that specific laptop is added to that file. The system creates a new file for every new client identifier that connects to the server. The server also stores some additional data. Specifically the external ip-address of the client and the time it connected and disconnected. The log of a connected session during the preparation phase is shown here:

Marian connected at 2013/05/14 10:02:58
130.89.170.205
(stud170205.mobiel.utwente.nl)
client: Backlog
client: Could no longer connect to server, 2013/05/14 10:03:13
client: Could not connect to remote host, info: 192.168.11.43 (marianne), 2013/05/14 10:03:13
client: Could not connect to remote host, info: 127.0.0.1 (marianne), 2013/05/14 10:02:29
client: Could not connect to remote host, info: 127.0.0.1 (marianne), 2013/05/14 10:02:29
client: End of backlog
client: 192.168.11.43 (marianne), 2013/05/14 10:03:13
client: 192.168.11.43 (marianne), 2013/05/14 10:03:18
Marian disconnected at 2013/05/14 10:03:04

Note here that every line sent by the client is prefaced with ‘client:’. This enables us to easily distinguish between the server and client timestamps. The logged in user is declared to be ‘Marian’, the client’s resolved remote hostname is ‘stud170205.mobiel.utwente.nl’ and the resolved local hostname is ‘marianne’. Windows XP does not allow the computer name to be equal to a username.

Requirement 4 is hard to quantify and a compromise was necessary. The police are still in the process of setting up the ‘playground’ for the digital volunteers. The only readily available solution was to use a home server. This particular home server is also used to play movies. Many people have access to the home server and so uptimes cannot be guaranteed. Experience shows that the home server is rarely shut down. However the server does not automatically log in to the default user after a reboot. As discussed before, the software does not run until a user logs in and this poses a threat to the tracker software’s uptime. Like the client, the server is packaged as a .jar and can only be closed in the Windows Task Manager.

2.3 Distribution

The distribution of the laptops into the environments that deal with stolen laptops is not trivial. With the information available to us there is no sure way of knowing which houses will be burgled in the near future. So having the laptops stolen in a real burglary is not feasible within the given time frame.

One alternative for distributing the laptops is to leave them unattended in areas which are known to have high crime rates. Here the assumption is made that the behavior of a person stealing a computer on the street is similar to that of a person stealing a computer in a burglary.

In a meeting with mister Henri Lucas, the Coordinator High Impact Crime team, three other alternatives for distributing the laptops were offered. In Mr. Lucas’ experience, stolen goods are sold as soon as possible after being acquired by the thieves. This means that there is a very low probability that the proposed solution yields any result at that stage. This provides the opportunity of simulating the theft. The laptops can be injected into the criminal environment at this first point of sale. These first points of sale are street deals at train stations, pawnshops and possibly charity shops.

Street deals at train stations are possibly the best known alternative for selling stolen goods. There is a low level of organization, if any, and so it is deemed unlikely that computer
experts will erase any tracking software or format and reinstall the laptop. On the other hand, it is a highly unpredictable environment and we do not have any experience with the involved practices. It would likely take a long time to use this alternative for distribution. It could also possibly lead to trouble with policemen who are not aware that this is an experiment, rather than illegal sale.

Pawnshops have a much higher level of organization. Most pawnshops are specialized towards certain goods. This means it’s likely that pawnshops that buy laptops have a certain level of expertise. This increases the chance that tracking software is erased or the laptop reinstalled, before it connects to the internet. Therefore pawnshops likely provide a lower boundary for the measured success rate. The protocol for selling a product here is much clearer than at a train station, because it is clear who the buyer will be.

Charity shops are similar to pawnshops in a number of respects. Both deal with second hand products and they cater to people who are looking for a bargain. Most products are sold ‘as is’, which is not necessarily the case with pawnshops. For the experiment this means the laptops will likely not be formatted or have their tracking software removed. Charity shops usually pay no money for the products they receive. This means that it is less likely that stolen goods are brought here. Charity shops are expected to be a good analog for poorly organized pawnshops.

2.4 Experiment Execution
After much preparation the laptops were distributed on May 13th and 14th 2013. Because of the limited time, the ease of selling to pawnshops was chosen as the method of distribution.

In Enschede four pawnshops were visited of which only ‘Used Products Enschede’ was interested in buying the Asus L5800C. The pawnshop ‘Cash Converters’ buys laptops that have a Windows OEM sticker, which the University’s laptops don’t have. Both pawnshops did mention that formatting the laptops is standard practice for them. Although these statements are considered to be an indication for the likely result, we felt experimental verification was still necessary. ‘Anne’ was sold to ‘Used Products Enschede’, which requires identification so that people can be tracked if they sell stolen goods.

Hengelo also has a ‘Used Products’ branch, and it is the only pawnshop in Hengelo that is advertised online. The Asus L3500Tp ‘Marianne’ was sold to them. Again identification was required. Additionally an agreement was signed to confirm the rightful transfer of ownership and to state that all installed software was legal. Together with the fact that the salesman did not state that the laptops would be formatted, this is an indication that the laptops are sold on without changing the software. The salesman also stated that this kind of laptop was expected to sell in 2-4 weeks.

Since all pawnshops in the immediate area had been used, the Asus L3500Tp ‘Yvonne’ was brought to the charity shop ‘Kringloop Enschede’. Again, a charity shop is expected to be a good analog for a poorly organized pawnshop. This is confirmed by their policy to sell the product without any modifications. The salesman stated that they expect the laptop to sell in 3-4 weeks.

After the distribution of the laptops, the tracking server is regularly checked to be running to ensure maximum uptime.

2.5 Results
On May 22nd ‘Marianne’ contacted the tracking server at 18:40. The complete backlog of 159 lines, 13KB, was transferred within an upper boundary of 2 seconds. The backlog shows the local ip-address to be ‘169.254.69.175’ for some time before changing to a regular internal ip-address ‘192.168.2.11’. An ip-address in the 169.254.x.x range indicates problems while configuring the network. After connecting at 18:40, the log shows some more network problems at 18:50 and it seems these were resolved after rebooting. Then from 19:00 onwards the laptop was continuously online, until shutting down at 20:36. After this ‘Marianne’ has not yet reconnected to the tracking server again.

Publicly available geolocation websites show that the external ip-address was located in Hengelo, where the laptop was sold. A whois lookup reveals that the internet service provider is KPN. To answer question 1, there is no further need for digging into the owner and location of the ‘Marianne’ laptop.

As of June 10th, the ‘Yvonne’ and ‘Anne’ laptops have not connected to the tracking server. It is expected that ‘Anne’ has been formatted and will never contact the tracking server. It is possible that ‘Yvonne’ has not yet been sold, and will contact the tracking server in the future.

3. IP RETRIEVAL
In the common case of a stolen laptop, the user will not have installed any dedicated tracking software. The police have specifically asked to research practical methods of retrieving the ip-address in such a case. Commonly installed programs that contact other devices through the internet provide opportunities to retrieve the device’s ip-address. Note however that the proposed solution of using already installed software to trace a laptop’s ip-address can always be defeated by reformattting the computer or uninstalling the specific software. The results in chapter 2.5 show though that the proposed solution is viable.

In the most common case, programs that contact the internet require a user to log-in after booting the computer. This paper focuses on these programs.

Programs that connect to the internet have either a client-server (Figure 1) or a peer-to-peer network architecture (Figure 2).

![Figure 1. Client-server architecture](image)

The client-server architecture is used in the tracking software described in chapter 2.2. Only the administrator of the server can practically find the ip-address of a given user. The servers of commonly used software are typically hosted by companies outside of the Netherlands. Additionally, these companies would need to actively monitor a user’s online-status. These complications result in a low probability of successfully using commonly installed client-server type software for retrieving ip-addresses, unless the software explicitly offers this.
The peer-to-peer network architecture eliminates the central server and allows devices to communicate directly with each other. This works very well for communication programs like Skype. Peer-to-peer software allows the police to wait until a stolen laptop connects to them and then log the corresponding ip-address.

Chapters 3.1 and 3.2 will treat practical methods for retrieving ip-addresses using Gmail and Skype, which are chosen because of their widespread adoption. Gmail is the world’s largest email provider with at least 287.9 million unique users per month [15]. In a blog Skype announced they have 280 million unique users per month [12].

3.1 Gmail

Google’s e-mail service Gmail is a client-server type web application that tracks the ip-address of its users. The last 10 sessions can easily be retrieved by clicking ‘details’ at the bottom-right of the inbox. This shows an overview with the browser, ip-address and timestamp per session. The timestamp shows either the time, if the session occurred in the last 24 hours, or the date. Optionally alerts can be turned on to notify the user of unusual activity.

Gmail was added to the experiment in chapter 2. It was set as the homepage and every laptop was signed in to its own user account. Gmail logged the session on the ‘Marianne’ account on May 22nd. This way both the data in the tracker server and the use of Gmail as a tracking tool could be validated. It must be noted however that Gmail likely isn’t everyone’s homepage. A user must then explicitly access Gmail in order to trigger its logging feature. Additionally the original owner must have left Gmail logged in and the cookie must still be valid.

3.2 Skype

The envisioned use of Skype for retrieving an ip-address is as follows. When a person comes to the police station to report a stolen device, the victim is asked whether or not they have Skype installed on the stolen device. If they do, they can then log in to their personal Skype account and add the police’s Skype account to their contacts. After this, the victim can’t use that Skype account again to ensure correct detection of the new, criminal user’s ip-address.

The Skype network architecture is a hybrid between the client-server and peer-to-peer topology [4]. A user logs in using the central Skype login server and registering itself to a super node. The central server is responsible for user authentication and insuring all usernames are unique. All other information is stored and maintained by the network. A computer running Skype can upgrade to super node, depending on the available computing power and network bandwidth [4]. The Skype network topology is illustrated in Figure 3.
A small experiment using netstat and Skype was performed with aid of two virtual machines. The only Skype contact that was added to each virtual machine was the other virtual machine. In this set-up some useful patterns emerged.

After starting, Skype connects to many ip-addresses, most likely in order to build its cache of super nodes. After some time, in the order of magnitude a couple minutes, Skype has only a small and stable number of connections. Skype does not have a connection to its contact on the other virtual machine at this point, regardless of whether or not that contact is online. However, after sending a chat message to the contact on the other machine, that machine’s ip-address (and only that machine’s ip-address) is added to the list of Skype’s connections. This only yields result when the contact is online when the chat message is sent.

When starting a call instead of sending a chat message, some additional ip-addresses are added to Skype’s active connections as well as the contact’s ip-address. It is currently unclear what these other connections are used for as well as what determines which peers Skype connects to. This makes starting a call unsuitable for determining the contact’s ip-address with our current understanding of Skype.

Skype provides an API that can be used to request the status (online/offline) of a contact and to send chat messages. An automated system can thus be created that combines the Skype API with netstat to determine a user’s ip-address.

3.2.2 Analyzing Skype Logs

Skype contains a feature that allows it to maintain an encrypted log file. This file is only readable for Skype personnel. Fortunately two versions of Skype exist in which this encryption, among other things, is removed. They are called ‘deobfuscated’ Skype. More specifically versions 5.5 and 5.9. The official version of the latter was released on December 4th 2012 [8]. The deobfuscated versions were not released by Skype and information on where they did come from seems only available on community forums and small technology news sites. The most likely scenario seems to be that the binaries were disassembled, the obfuscation code removed and then recompiled. This is allowed for interoperability purposes by European Union law [6]. We feel that all requirements of this law are met to legally use the deobfuscated version 5.9 of Skype for interoperability with an automated ip-tracker for stolen laptops.

We confirm that the deobfuscated version 5.9 is a working Skype client which writes a log in clear text. It appears that Skype stores the external ip-address of its contacts in this log file, even if no messages are sent between the contacts. One section containing the relevant contact information of the user ‘zangief7057’, which was used in the experiment, is shown here:

```
09:44:47.107 T#2948 PresenceManager: &w-,~zangief7057
09:44:47.107 T#2948 PresenceManager: &a-noticing-zangief7057-0x7bf9731d9e825b3-s=64.4.23.154:40029-r130.89.167.141:53900-1192.168.32.131:53900-68b102a9-2f93a75e
09:44:47.107 T#2948 PresenceManager: &w-,~zangief7057-7bf9731-d9e825b3-2-2
```

4. IP-ADDRESS TO PHYSICAL ADDRESS

After finding the ip-address of a stolen laptop, the physical address has to be determined. The Dutch police have access to the ‘Centraal Informatiepunt Onderzoek Telecomunicatie’ CIOT by way of the CIOT Information System (CIS). This provides a 91% nationwide hit rate [11].

The Enschede police team of forensic ict specialists have stated that the police have successfully used 126na Sv to demand user data for an ip-address from an internet service provider in the past. They stated that a custom approach per case was always necessary though.

126na Sv is the formal notation for article 126na of the Dutch penal code ‘Wetboek van Strafvordering’. A translation of 126na Sv member 1 is provided. It must be noted that the original Dutch text is always leading. However we feel this is a sufficiently accurate translation for the purpose of this paper.

126na Sv, member 1 translation:

1 In case of suspicion of a crime, a public official tasked with the investigation can, in aid of the investigation, subpoena the name, address, postcode, area of residence, number and type of service of a user of a communication service.

In 2011 25% of all CIS requests were based on 126na Sv and a further 74% on 126n Sv [11]. Burglary, theft and fencing are crimes that seem to qualify for 126na Sv. Article 126n has stricter requirements considering the type of crime and only seems to apply to burglary, theft or fencing if the suspect has been convicted in the last five years, according to articles 67 and 67a Sv.

5. CONCLUSION

The goal of this paper is to evaluate the viability of the proposed solution for retrieving stolen laptops. The posed sub-questions will be evaluated and then related to the main research question.

The chance of a stolen computer being connected to the internet must be expressed qualitatively due to the small sample size of the experiment in chapter 2. Out of the three laptops one reconnected to the tracking server. From this we must conclude that there is a real chance of a stolen laptop connecting to the internet. The combination of the salesmen’s statements on whether or not the laptops are formatted and the result of our experiment, are an indication of the effectiveness of the proposed solution in Enschede and Hengelo respectively.
In chapter 3 a number of practical methods were presented for retrieving ip-addresses using the commonly used software Gmail and Skype. Gmail has a built in feature to show the ip-addresses of the last 10 sessions. Skype can be used by either monitoring network connections when sending chat messages or using the deobfuscated logs. General observations regarding the feasibility of using different network architectures for ip-address retrieval were also offered. This may aide the police in selecting additional software for tracking ip-addresses.

An ip-address is matched to a physical address using CIS. In theory this is straightforward by invoking 126na Sv. In practice however, a custom approach per case has always been necessary.

Considering the answers to the sub questions, we feel that the proposed solution for retrieving stolen laptops is viable. Especially bearing in mind that even solving 1% of burglary cases using this solution is significant in comparison to the 12% that were solved last year.

6. FURTHER RESEARCH
Due to a number of different constraints, some topics were less thoroughly researched than they deserve.

Our main recommendation is to execute the experiment in chapter 2 on a larger scale. Specifically using more laptops as well as other devices will allow statistical analysis. Using more diverse methods of distribution will provide insight into less well organized channels.

Our second recommendation is to further research the use of unencrypted Skype logs. Practically the exact formatting of the log to extract the contact’s ip-addresses is required. Furthermore the exact legal state of the software must be researched by IT law experts. We feel that the European law allows the use of the deobfuscated Skype client, but the police will require confirmation before being allowed to use it.

Our last recommendation is for the police to research how best to standardize the use of 126na Sv since it is not feasible have a per case custom approach. Additionally a legal expert should verify the assertions made in Chapter 4.

7. ACKNOWLEDGMENTS
Our thanks go to the Twente police and to dr. ir. Nirvana Meratnia for facilitating this study, to ICTS for providing the laptops and to prof. dr. Pieter Hartel and the others who provided valuable input and feedback.

8. GLOSSARY
API Application programming interface, this allows the interaction between different programs.
Cache An easily accessible collection of data
CPU A computer’s central processing unit
Geolocation A service which matches an ip-address to a physical location. The accuracy is usually in the order of towns.
Ip-address A unique number that identifies a device on the internet
JAR A packaged Java program
Java A programming language and runtime environment. Computer Science bachelor students at Twente University are experienced with making network programs in this language.

Pawnshop A shop where items can be sold or temporarily pawned. In Dutch: pandjeshuis
RAM A computer’s working set memory
SSH Secure Shell, a protocol for secure communication between two networked devices
Sv Shorthand for the Dutch penal code ‘Wetboek van Strafvordering’
VNC A program for graphically accessing a remote computer
Whois A service to request administrative information on the owner of an ip-address.

9. REFERENCES


