INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Ciber Segurança Forense

MEIC / METI 2018-2019 – 1º Semestre

Relatório de Forense Digital

Autores: Inês Paiva (77926), João Meira (82014), Zé Eduardo (82069)

1 Objetivos da investigação

O objetivo desta investigação é procurar provas que possam levar à resolução do caso "Mole Affair". Para alcançar isto, irá ser necessário examinar um pequeno número de ficheiros retirados da drive de John Mole. A examinação destes ficheiros irá ser efetuada utilizando técnicas de análise forense de ficheiros e esteganografia e, para ter a certeza de que os ficheiros não são danificados durante o processo, uma distribuição Kali Linux numa máquina virtual forense irá ser usada (Kali Linux – Forensics Mode).

2 Artefactos para análise

De forma a contribuir para a investigação, foi juntada uma equipa de ciber segurança forense para investigar os artefactos digitais na drive do suspeito e procurar por potenciais provas de espionagem industrial. Os artefactos digitais obtidos foram os seguintes: quatro ficheiros .png e um ficheiro .bmp, imagens normais à primeira vista; um ficheiro .txt acerca da história de Munique; um ficheiro .py chamado "compress.py" que agarra na informação de um ficheiro de texto e a esconde numa imagem, utilizando um algoritmo de esteganografia chamado "LSB", tal como detalhamos mais à frente; por último, existe um ficheiro .zip que está protegido com *password*.

Ficheiro	**Valor MD5**
munich.txt	c6596b360ac97889c4f2d68ba6787f92
compress.py	72eab63334dcd0f73418e32999b71f05
cathedral.png	55fd5b1d42072955e15769b55a390400
oktoberfest.png	deb345aea6cdb82ca4636c0811c292df
street.png	f1ea1beaa6a838d16b4d457c6fe68fd0
wursten.png	13c85b20b6b1e481a32700f26818333e
snow.bmp	a6e56c4d34d9a541b622b74c954c3fc9
online_banking.zip	b3baa737b818db4f52a681f0cf8d440c

3 Provas a procurar

Dado que, à primeira vista, as imagens são ficheiros normais, terão de ser utilizados vários métodos forenses para tentar descobrir provas escondidas que sejam relevantes para o caso. Inicialmente, foi analisado o ficheiro de texto que poderá ter informação que ajude a perceber por onde deve começar esta investigação. Posteriormente, foram analisados os ficheiros de imagem, usando comandos como "strings" e "hexdump". De seguida, procedeu-se à tentativa de aceder aos ficheiros escondidos pelo ficheiro zip. Sendo que este último estava protegido por *password*, a nossa hipótese inicial é que haveria de entre os ficheiros um dicionário que nos fosse permitir encontrar esta *password*. Para além disto, e após termos assumido inicialmente que o "compress.py" serviu para criar o ficheiro online_banking.zip com *password*, analisámos melhor o ficheiro compress.py utilizando o método de "reverse engineering" e criámos uma nova hipótese: que alguém utilizou este programa para esconder dados nos ficheiros .png utilizando o algoritmo de esteganografia LSB. Mais tarde e, finalmente, tendo já descoberto uma *password* escondida num ficheiro do .zip, assumimos desde logo que esta *password* iria ser utilizada para extrair a possível informação escondida nos ficheiros .png. Este processo é descrito em mais detalhe na secção 4 e, ainda em mais detalhe, num ficheiro anexado ao qual chamámos de "trial_and_error.txt", que não é utilizado aqui na íntegra devido à sua falta de organização (foi sendo feito à medida que avançávamos no trabalho).

4 Detalhes da examinação

A examinação resultou numa série de segredos encontrados dentro dos ficheiros. Utilizando diversos métodos de análise forense dependendo do ficheiro e das suspeitas que tínhamos acerca dele, conseguiu-se encontrar as seguintes provas forenses:

- Snow.bmp: esta foi a primeira descoberta. Usando o comando "strings" foi possível descobrir que dentro deste ficheiro de imagem tinha sido escondida uma mensagem em alemão. Através do tradutor chegou-se à conclusão de que a mensagem escondida era uma lista dos cinco ficheiros que o suspeito estaria a enviar (apesar de estarem apenas listado 4, o criminoso afirmou que estava a enviar 5 ficheiros). A mensagem mencionada foi traduzida para inglês, e eis a sua tradução: "*I send you five files: (1) Drone A Plans, (2) Drone B Plans, (3) Technical Specifications, (4) Passwords of DroneX File Servers"*;
- Munich.txt: existe a suspeita de que este ficheiro possa ser usado como dicionário para descobrir a *password* que protege o ficheiro .zip. No entanto, para ser um dicionário é necessário que só exista uma palavra por linha, o que não acontece, pelo que foi criada uma cópia do ficheiro (à qual se chamou munich-copy.txt) e posteriormente foi corrido o comando "sed -i 's/\ /\n/g' munich-copy.txt" para separar cada palavra numa linha diferente e assim tornar este novo ficheiro num dicionário que possa ser utilizado;
- online_banking.zip: este ficheiro está protegido por *password*, pelo que, para extrair os ficheiros deste .zip, foi necessário usar o ficheiro munich-copy.txt como dicionário. Utiliza-se o então o comando "fcrackzip -u -D -p munich-copy.txt online_banking.zip", que consegue obter a palavra chave que era necessária para abrir o zip, e que, neste caso, é "Stadelheim". Utilizando a *password* encontrada para abrir o ficheiro zip obtém-se dois novos ficheiros: drone-A.bmp e online_banking.docx;
- online_banking.docx: sabendo que os ficheiros .docx são, no fundo, um ficheiro .zip, cria-se uma cópia deste ficheiro e faz-se "unzip", o que resulta numa estrutura de ficheiros típica de um documento "word" normal. Após isto, abre-se o ficheiro .xml correspondente ao conteúdo do ficheiro Word, denominado "document.xml". Abrindo o ficheiro encontrado no programa sublime (que facilita a leitura do .xml), é encontrado algo suspeito: "Password: 51782";
- drone-A.bmp: inicialmente não é possível abrir este ficheiro. Ao correr o comando "strings" verifica-se que existe um "IEND" no final do ficheiro. Sendo que esta "*keyword*" é utilizada nos ficheiros .png suspeita-se que o ficheiro em analise possa ser na realidade um .png em vez de ser um .bmp. Ao correr o comando "hexdump" num ficheiro .png pode-se verificar que o "Magic number" deste tipo de ficheiros ocupava os primeiros 8 bytes, e ao correr o mesmo comando no drone-A.bmp esses 8 primeiros bytes

coincidem parcialmente com esse mesmo número. Sendo assim, é utilizado o comando "hexeditor" para editar os primeiros 8 bytes e guardar o ficheiro novo com o nome "drone-A-corrected.png". O novo ficheiro já é possível de abrir e parecem ser os planos de um *drone*, o A.

- compress.py: na realidade, o ficheiro "compress.py" é um ficheiro python 2.7 "byte-compiled", tal como referido pelo comando file quando corrido sobre este ficheiro. Como tal, mudámos a extensão deste ficheiro para "compress.pyc" e fomos à procura de um programa que faça a descompilação de código python compilado, tendo sido escolhida a biblioteca uncompyle6 para tal efeito. Depois de feita a descompilação e termos analisado o código, verificámos que este ficheiro não tinha a finalidade que tínhamos suposto inicialmente, a de compilar e encriptar ficheiros para zip, mas sim a de esconder informação nos 2 bits menos significativos de cada byte das cores RGB de uma imagem no formato .png após lhe ser dada uma *password* que determina o offset para iniciar a compressão. É de mencionar que cada pixel é composto por 4 bytes dado que o esquema de cores é RGBA. No entanto, este script não esconde informação no byte correspondente à componente "Alpha" de cada pixel. Após um longo processo de engenharia inversa e uma análise detalhada do código, para este fim, partiu-se à criação de um programa que com base nas informações acima extrai a informação possivelmente encriptada nos ficheiros .png que foram encontrados na drive do suspeito. O processo de compressão de dados é explicado mais detalhadamente no ficheiro "trial_and_error.txt".

  - Com isto foi completado o script "extract.py", cuja funcionalidade recupera a informação encriptada nos últimos 2 bits menos significativos de cada byte das componentes RGB, de cada pixel (RGBA), recebendo a imagem .png e a *password* como inputs, e devolvendo como output os bytes desencriptados convertidos para ASCII. Assim, este script em python foi aplicado a todas as imagens .png na drive em junção com a *password* 51782 obtida no documento online_banking.docx. É de mencionar que antes de o fazer foi verificado o md5 de cada ficheiro, verificando-se que se mantêm iguais aos obtidos no início da investigação. No entanto, tal como explicado mais detalhadamente no ficheiro "trial_and_error.txt", o uso desta *password* na extração estava a causar com que os primeiros 3 bytes dos ficheiros escondidos não fossem revelados, uma vez que o "displacement" seria 3 (51782 mod 13). Tentámos, portanto, correr o programa extract.py com a *password default* (password = 0). Após corrido o programa deste modo, obtiveram-se os seguintes segredos:

  - wursten.png: esta imagem que inicialmente parecia inocente, continha escondida a seguinte "string" de texto, revelada através do comando "cat" ou "strings" ao output do programa:

  "ACCESS PASSWORDS FOR DRONEX SERVERS

  SERVER 1: Sr!_01llxt

  SERVER 2: p_GEtKl4dA"

  Podemos observar as *password*s para os servidores 1 e 2 da própria companhia, a *DroneX*. Esta informação refere-se muito provavelmente ao índice 4 do que o responsável disse que ia enviar (as *password*s dos servidores) de acordo com o que obtemos acima no ficheiro 'snow.bmp'.

  Este passo é diretamente reproduzível utilizando o comando: "./extract.py ../wursten.png 0 > wursten-output.txt"

  - oktoberfest.png: correu-se o script de extração nesta imagem e redirecionámos o output para o ficheiro 'oktoberfest-output.txt'. Correu-se o comando file neste ficheiro, o que revelou que este era na realidade um ficheiro .png. Procedemos então à abertura deste ficheiro, o que revelou a imagem dos planos de um *drone* que era diferente da imagem do *drone* anterior. Pensamos que esta imagem sejam os planos do *drone* B, tal como descrito no índice 2 da lista dos ficheiros encontrada no 'snow.bmp'.

  Este passo é diretamente reproduzível utilizando o comando: "./extract.py ../oktoberfest.png 0 > oktoberfest-output.png"

- street.png: correu-se o script de extração nesta imagem também, tendo obtido uma outra imagem .png. Após visualizarmos a imagem, pensámos na hipótese de que esta imagem poderia ela também ter dados escondidos (*inception*). Voltámos, portanto, a correr o script com a mesma *password*, mas desta vez na imagem obtida. Corremos então o programa file de novo no output, tendo recebido a informação de que era um ficheiro "UTF-8 Unicode text", pelo que corremos o comando "cat" no mesmo. Descobrimos o seguinte texto:

"OPERATING SPECIFICATIONS

DRONE A

Operational Altitude: 5,000 – 12,000ft WGS, no lift loss

(FAA limited at this time)

Sensor agnostic system

Full autonomous flying, terrain following, target tracking, real time GPS waypoints

Dual Mode – Wireless and Laptop

Automated return home

Max / min altitude setting

Payload up to 15lbs

Can fly in 30mph crosswinds

Take off and land in small areas

Flight Management Integration

Auto Takeoff, flight, and Landing

Waypoints – 10cm accuracy terrain following

Remote camera settings

GPS integration

Real time downlink data

Flight Time

Gas Powered: 60 – 90 minutes w / 10 – 15lb payload

Electric Powered: 25 – 45 minutes w / 10 – 15lb payload

DRONE B

Sensor agnostic system

Full autonomous flying, target tracking, real time GPS waypoints

Dual Mode – Wireless and Laptop

Automated return home

Max / min altitude setting

Payload up to 1lb

Take off and land in small areas

Flight Management Integration

Auto Takeoff, flight, and Landing

Waypoints – 5m accuracy

Remote camera settings

GPS integration

Real time downlink data

Flight Time

Electric Powered: 20 minutes w / 1lb payload

”

Encontrámos as especificações técnicas do Drone-A e Drone-B, ou seja, o índice 3 da lista.

Este passo é diretamente reproduzível utilizando o comando “./extract.py ../street.png 0 > street-output.png ”, seguido de “./extract.py street-output.png 0 > street-output-output.txt”

- catedral.png: corremos o script de extração neste ficheiro também e, não tendo encontrado nada relevante, utilizámos diversas outras técnicas forenses para tentar pelo menos descobrir alguma pista que nos levasse a descobrir algum tipo de ficheiro específico aqui oculto. Para isto, fizemos um estudo através do uso do “hexeditor” e “hexdump” e fizemos um “*search*” de alguns Magic Numbers neste, não tendo encontrado absolutamente nada relevante. Para além disto, alterámos o script de forma a utilizar a *password* 51782 na íntegra (isto é, utilizar a *password* “51782” e não a *password* “51782 mod 13”). Isto não nos foi útil pois não encontrámos nada, e, portanto, ficámos sem saber o que mais fazer para descobrir possível informação escondida neste ficheiro.

Por esta altura, faltava-nos um suposto “quinto” ficheiro para encontrar, apesar da lista ter apenas quatro, e descobrir para que servia então a *password* “51782”. No entanto, depois de termos discutido este assunto entre nós, concordámos que o criminoso se deve ter enganado e queria dizer “quatro ficheiros”. E quanto à *password*, esta pode ser a *password* da sua conta num banco online.

Ficheiro	**Valor MD5**
munich-copy.txt	902627e46c0a72f71d093645d0713dcc
online_banking.docx	b70702822417bd39a7997a0f8c73941f
drone-A.bmp	05029f0ae6af62ca3350f5b094584b22
drone-A-corrected.png	d99f500968d444b5e0a1c9fd1dd69274
compress.py	c641e7e24abda594f0b9ba70f4e2efd6
compress.pyc	72eab63334dcd0f73418e32999b71f05
black.png	4937c842542d2acaf94b1299151ee297
cathedral-output.txt	aa8f5fcf6d8c06c681050a541440d546
mensagem.txt	9a485b914a33c4ebce24fefd7e5f2358
oktoberfest-corrected.png	fe7920de632af2b4716cecd0cfc2c66b
oktoberfest-output.png	5217527b444f5dc5f44fd9c8c8663aad

street-output.png	7e6d51a4997704a832d3866405c02792
street-output-output.txt	dfb4cc39c2a873eca69403ee1e532e25
white.png	e68fd749681d02e583c364f690f7e4a5
white.png-stego.png	1788880a7552fe7371e24f3816507933
wursten-output.txt	c5c7d9afc5fd46d7d4e44404b82bb660
RGBA_convert.py	0b353a6f3450c93de1a0a15547f35b67
extract.py	691d02e252c440846dbee2169592e239

Os ficheiros cujo nome contém "output" são derivados dos ficheiros originais, extraídos utilizando o programa "extract.py". Os ficheiros não mencionados anteriormente são ficheiros que usámos para testes apenas, nomeadamente "black.png", "mensagem.txt", "oktoberfest-corrected.png", "white.png", "white.png-stego.png" e "RGBA_convert.py".

5 Resultados da análise

Temos razões para acreditar que na vinda da Alemanha, os dados estivessem já ocultados, como uma medida de prevenção contra uma potencial investigação à drive. Apesar disto, não podemos concluir que foi John que escondeu esta informação, podemos apenas afirmar que ele a tinha na altura da sua apreensão.

Note-se que todas as fotos que John carregava consigo na sua drive e o texto sobre a cidade de Munique pareciam, à primeira vista, inocentes. No entanto, o que nos fez suspeitar desde logo que a drive continha, realmente, informação confidencial da empresa escondida, foi a existência de um ficheiro .zip protegido por *password* e de um script python chamado "compress.py" junto dos outros ficheiros.

As imagens que John Mole trazia consigo, como mencionado acima, pareciam típicas de alguém que fez uma viagem à Alemanha; notemos, por exemplo, a foto das salsichas, a de uma catedral ou o texto com informações retiradas da Wikipédia sobre a vasta história da cidade de Munique. O criminoso, ao utilizar estes aparentemente inocentes ficheiros, pretendia claramente esconder as suas verdadeiras intenções das autoridades envolvidas. Provavelmente, após ler o texto da Wikipédia sobre a cidade, decidiu utilizar a palavra "Stadelheim" como *password* para o ficheiro .zip e, como tal, manter o ficheiro ".txt" por perto para, caso se esquecesse da password, lhe bastasse apenas ir ao capítulo correto para se relembrar como se soletrava. Esta palavra foi retirada da parte do capítulo sobre o final da primeira guerra mundial e a revolução nacional socialista alemã que retratava a infame ocasião conhecida como "A Noite das Facas Longas", na qual o governo alemão purgou múltiplas fações do partido nazi, incluindo as forças de assalto, também conhecidos como os camisas castanhas, e o seu líder Ernest Röm na prisão de "Stadelheim". Provavelmente foi escolhida como *password* por ser um edifício tão negro (num dos períodos mais negros da história) e, como tal, inesquecível.

No entanto o facto do suspeito ter no meio de todas as imagens o ficheiro ".zip" que era protegido pela *password* anterior faz-nos sem dúvida ponderar que este tenha escondido os dados à pressa, para além do facto de não ter renomeado o nome do ficheiro "drone-A.bmp", tornando óbvio do que o ficheiro trata. Adicionalmente, o próprio título "*online banking*" sem dúvida que é uma boa desculpa para um ficheiro zip com *password*, dado que se tratando de informação bancária esconderíamos os nossos extratos ou conta com uma *password*, e ninguém nos pediria para extrair os ficheiros do zip sem um motivo forte. A existência de uma imagem chamada "catedral.png" sem informação aparente suporta esta teoria, uma vez que poderia ser nesta imagem que o criminoso iria esconder a informação do *drone* A; apenas não o terá feito por não ter tido tempo para o fazer.

Por último, este ficheiro ".zip" tinha também a *password* "51782" num ficheiro ".docx". Após a investigação concordámos que em nada ajudava na extração de informação dos ficheiros ".png" ou de qualquer outro ficheiro da drive. No entanto, a mera existência desta *password* e o facto de não ser utilizada, em junção com o próprio

nome do ficheiro "online_banking.docx", faz-nos considerar que esta poderia ser a *password* que este usava para o seu serviço bancário online, que acreditamos que seja algo recente para o criminoso dado que, se não o fosse, este não a esconderia num ficheiro por medo de se esquecer. Como tal, a nossa teoria é que decidiu escondê-la, também, por detrás do ficheiro .zip. Possivelmente, esta será a conta bancária online na qual o criminoso receberia os pagamentos por vender os segredos da empresa *DroneX*.

Na foto "Snow.bmp", a nossa teoria é que o criminoso decidiu esconder a informação textual que foi enviada para os seus "clientes" num ficheiro BMP devido à facilidade de esconder informações neste tipo de ficheiro. Dado que os ficheiros deste tipo definem os bytes que a imagem ocupa logo no seu cabeçalho, torna-se possível esconder o texto no final sem corromper a imagem original. Sendo apenas uma mensagem curta e dado que o script python "compress.py" servia para esconder informações em imagens .png, o criminoso não achou que fosse necessário esconder esta mensagem em nenhum dos ficheiros .png e, como tal, usou esta imagem .bmp para esconder a breve mensagem. Além do mais, esta mensagem estava em alemão. A explicação mais provável para isto é que os seus clientes sejam da Alemanha, o país que John Mole foi alegadamente visitar. Como bónus, o criminoso pode ter achado que seria ainda uma outra forma de dificultar o trabalho de análise forense de alguém do seu país, uma vez que a probabilidade de saberem alemão seria reduzida.

Como mencionamos anteriormente, achamos que o ficheiro "drone-A.bmp" estava propositadamente com uma extensão e "Magic Number" errados. Sem dúvida que foi uma tentativa de ocultar de 2 maneiras distintas a informação que este ficheiro escondia. Ainda para mais, o próprio ficheiro estava protegido por detrás de um ficheiro .zip com *password*.

O programa compress.py, no entanto, é talvez um dos ficheiros mais suspeitos à partida. Por que razão é que alguém teria um programa python com as fotos de férias? Este programa, à semelhança da imagem drone-A.bmp, tinha a sua extensão alterada para um formato errado. Neste caso, era um ficheiro de código python 2.7 compilado, mas tinha a extensão de um ficheiro python normal. Novamente, assumimos que foi uma forma preguiçosa (ou feita à pressa) e pouco inteligente do criminoso de tentar esconder o que o ficheiro realmente é.

Finalmente, temos as outras três imagens, as quais tinham informações confidenciais da empresa de John. Estas imagens são a "wursten.png", a "street.png" e a "oktoberfest.png". Todas estas imagens tinham informação escondida pela mesma funcionalidade do ficheiro "compress.py". Provavelmente, a imagem "cathedral.png" iria ter a informação referente ao Drone-A lá escondida também, no entanto, o criminoso pode não ter tido tempo de o fazer. Na imagem "wursten.png" este escondeu a informação referente às *password*s dos servidores de ficheiros da *DroneX*. Na imagem "oktoberfest.png" foi escondida a imagem do Drone-B, tal como mencionado na informação obtida em snow.bmp. A forma como esta informação estava escondida em imagens diferentes fez-nos pensar na existência de alguma relação. Finalmente a imagem "street.png" foi a que teve mais cuidado no tratamento por parte do criminoso. Este usou o programa para lá encriptar outra imagem, a de um castelo bávaro. Segundo a nossa pesquisa, o castelo em causa é o castelo de *Neuschwanstein*. Mas por que razão é que alguém se daria ao trabalho de encriptar uma imagem dentro de outra imagem? Claramente que este tinha um grande apreço por esta foto, ou então pela informação nela escondida. Foi exatamente isso que nos levou a correr o programa de novo na imagem obtida, o que nos deu acesso às especificações técnicas de cada um dos *drones*, A e B. Foi o maior texto que até agora encontrámos e, ao mesmo tempo, o texto que mais parece importante para o desenvolvimento dos *drones*. Sem dúvida que esta informação era a que o criminoso melhor queria esconder.

6 Conclusões

Após todo o processo de investigação e a descoberta de provas incriminatórias na drive do suspeito, podemos concluir que a viagem que John Mole fez até à Alemanha provavelmente não era uma viagem de lazer, mas sim, uma viagem de negócios ilegal na qual foi tentar vender os segredos da empresa a companhias concorrentes sediadas no país. A presença de mensagens em alemão com informações secretas sobre os *drones* da empresa são provas fundamentais que suportam esta teoria. No entanto, é importante mencionar que uma prova mais definitiva seria a confirmação da nossa hipótese de que a conta de *online banking* de John Mole é, de facto,

protegida pela *password* que encontrámos e, também, encontrar algum tipo de pagamento suspeito na sua conta. Para tal, é definido em baixo o possível trabalho futuro a ter em conta, dado o contexto.

Trabalho futuro: a existência de uma suposta *password* de uma conta bancária online é uma prova adicional que nos poderá fazer não só encontrar o comprador destes segredos, mas também confirmar que esta conta pertence, de facto, a John Mole. Caso pertença, será uma prova sólida de como John Mole é o vendedor da informação confidencial da *DroneX*. Por conseguinte, o próximo passo nesta investigação seria apresentar estas provas a um Juiz de forma a poder obter um mandato que nos dê acesso legal ao seu computador, de forma a poder procurar no seu histórico do *browser* por algum *website* de *online banking*, no qual se tentaria aceder à conta bancária de *John Mole* com esta *password*. Poder-se-ia então, desta forma, confirmar que a conta de John Mole era a conta cuja *password* estava presente no documento Word, caso conseguíssemos encontrar esta conta (poderíamos, por exemplo, procurar na cache do *browser* no qual ele navegou para o website para tentar descobrir que conta é esta ou, em último recurso, pedir aos donos do *website* que colaborassem na investigação). Adicionalmente, poder-se-ia confirmar se John Mole tinha recebido algum pagamento suspeito, de modo a poder descobrir o possível comprador que estaria também ele a cometer um crime grave. Poderíamos também procurar por websites de empresas alemães no histórico do browser, tal como fazer um script para fazer "GET requests" aos sites todos que John acedeu e detetar certas palavras alemães relacionadas com *banking*. Poderíamos também procurar nas redes sociais, telemóvel e e-mails de John para procurar por mais provas relacionadas com este caso.

Assinado por:

Inês Barral Paiva, nº 77926

João Pedro Meira, nº 82014

José Eduardo Brás, nº 82069

No dia: 24-10-2018

Apêndice:

Iremos anexar, junto com este documento PDF, um ficheiro ZIP com o nosso ambiente de trabalho para este projeto. Referimos a existência de uma diretoria denominada "auxiliary_files", a qual contém todos os ficheiros auxiliares e ficheiros úteis de teste, e uma diretoria denominada "evidence" na qual estão todas as provas encontradas. Estes ficheiros já foram todos referidos anteriormente na secção 2 e na secção 4, onde se podem verificar os respetivos valores MD5. Queremos também enfatizar que, de entre os ficheiros auxiliares, está um ficheiro denominado "trial_and_error.txt" com uma descrição detalhada de todos os passos que realizámos, à medida que os fomos realizando. Apesar de este ficheiro não conter uma organização tão boa quanto este relatório, serviu como um guião para fazer o relatório uma vez que o fomos escrevendo à medida que progredíamos com a nossa investigação. Enfatizamos também que, de entre os ficheiros, está o programa "compress.py" descompilado e o programa utilizado para extrair a informação das imagens .png ("extract.py").

A outra diretoria, denominada "csf-lab1-artifacts", contém os artefactos digitais originais que estavam na drive do suspeito, John Mole.