INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Forensics Cyber Security

MEIC / METI 2018-2019 – 1st Semester

Digital Forensics Report

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

1 Can you determine how the malware has taken over Sally's computer?

Analisando o disco com as ferramentas do TSK e utilizando um programa que foi criado por nós que faz icat de todos os ficheiros encontrados recursivamente para as diretorias listadas do disco ("parse_disk.py), é possível reconstruir o file-system da Sally. Para determinar como o malware conseguiu entrar no sistema da Sally fomos à procura de informação relevante analisando a pasta "/home/sally/". Após análise dos conteúdos desta diretoria, encontrámos em "/home/sally/.cache/thunderbird/" os ficheiros correspondentes ao perfil Thunderbird da Sally, o que nos permitiu aceder aos seus e-mails. Para ler os seus e-mails, instalámos a aplicação Thunderbird na nossa máquina virtual e copiámos os ficheiros da diretoria r9ufkrkl.default da Sally onde se encontra o seu perfil para a nossa diretoria correspondente. Tendo acesso à sua caixa de correio, reparámos numa mensagem que tinha como assunto "Important Security Update" e tinha como remetente a conta jason_halloween@protonmail.com, o que nos pareceu algo suspeito uma vez que fala numa atualização de segurança e que o e-mail utilizado é o mesmo e-mail que apareceu no pop-up do ransomware. Dentro do e-mail, o individuo explicava que a rede do campus tinha sido atacada e que a era necessário que a Sally atualizasse as suas definições correndo o programa "main" que vinha em anexo. Este e-mail estava assinado por "Sid Wilkes" que, alegadamente, pertence ao departamento de tecnologias de informação. Suspeitamos que a assinatura tenha sido forjada, ou seja, que o verdadeiro remetente não seja a pessoa que consta na assinatura do e-mail. Cremos que o malware tenha conseguido acesso ao computador da Sally a partir do programa que vinha em anexo neste e-mail. As razões que temos para acreditar nisto são que o e-mail do remetente é equivalente ao e-mail que consta na janela de pop-up que o malware apresentou à Sally. Para além disto, o nome deste programa é "main" e, de entre os processos analisados, foi este que criou uma pasta temporária e fez operações de criptografia nesta, de acordo com os outputs do comando volatility (linux_pslist, linux_proc_maps e linux_elfs) anexados. Cremos, portanto, que a Sally, tendo sigo coagida a achar que correr o programa "main" a protegeria da falha de segurança (possivelmente falsificada) reportada no e-mail de má índole, descarregou o programa e correu-o, tendo comprometido desta forma a segurança da sua máquina. Assim, após iniciar a execução, o programa malicioso recorreu a certas bibliotecas de criptografia (libcrypto.so.1.0.0, libssl.so.1.0.0) e encriptou recursivamente todos os ficheiros encontrados na '/home/sally/Documents' utilizando como base a diretoria '/tmp/_MEIlXS6RU/', na qual nós suspeitamos que fez a maioria das operações malignas.

2 Can you recover Sally's original files? If you do not succeed at retrieving the original files, can you at least extract some of its fragments?

Quando começámos a análise do computador da Sally decidimo-nos focar primeiramente na memória pois esta podia ter qualquer indício dos processos que foram usados para cifrar os ficheiros. Para analisar a memória recorremos ao uso do programa volatility. Os comandos que nos deram os outputs mais relevantes para a análise da memória foram os comandos "linux_pslist" e "linux_pstree". Após obtidos estes dois outputs (que se encontram na pasta "auxiliary_items/volatility_output/") começamos por pensar que os processos "crypto" e "encryptfs" fossem os suspeitos em causa, pois estes estiveram em execução e o seu nome explicita claramente a sua função como processos de encriptação do sistema operativo Linux. No entanto o que nos chamou mais a atenção foi a existência de dois processos chamados main no final da lista de processos "linux_pslist.txt". Ambos estes processos, devido à sua nomenclatura, pareceram algo suspeitos, pois main é geralmente o nome que se dá à função principal de um programa e, também comparando com a nomenclatura de todos os processos linux encontrados, este parecia algo deslocado. Para além disto, este era o nome do programa entregue na caixa de correio da Sally e, teoricamente, descarregado pela mesma (a existência de um ficheiro "main" nos Downloads da Sally e o timeline próximo destes dois eventos correlaciona-os fortemente um ao outro). Adicionalmente, este processo main fazia um "fork" de si mesmo, sendo evidência disto a existência de outro processo main com o Ppid igual ao Pid do processo main anterior (evidente no ficheiro "linux_pslist.txt" presente na pasta "auxiliary_items"). Ou seja, claramente o processo main inicialmente fazia um "fork" de si mesmo criando um processo filho para executar outras operações. Esta situação tendo-nos chamado a atenção fomos investigar e monitorizar a existência de todos os processos que eram filhos de outro processo na lista obtida. Para obter informações sobre os processos em execução na memória corremos o comando "linux_proc_maps" do volatility de forma a obter todos os detalhes de cada processo na memória, inclusive as bibliotecas que eles utilizam. O ficheiro de output obtido foi guardado em "linux_proc_maps.txt" que, similarmente aos anteriores, se encontra no diretório "auxiliary_items/volatility_output/". Após analisar o conteúdo deste ficheiro, nomeadamente tendo atenção aos processos que considerámos problemáticos anteriormente (incluindo os processos que eram filhos de outro na lista), encontrámos novamente o nosso suspeito "main". Analisando em detalhe o seu comportamento verificámos que este chamava diversas vezes bibliotecas de encriptação, inclusive, chegando a importar para uma pasta criada no diretório dos ficheiros temporários "/tmp/_MEIlXS6RU/" bibliotecas de encriptação como "_raw_aes.so" e "_raw_ocb.so" copiados para "/tmp/_MEIlXS6RU/Crypto/Cipher/", correspondendo respetivamente à cifra de encriptação de chave simétrica AES (Advanced Encryption Standard) e ao modo de encriptação por blocos OCB (Offset CodeBook Mode). Noutras operações mais abaixo este processo movia outras bibliotecas temporárias com outros modos de encriptação em cifra por blocos como OFB, CBC, EBC, CTR. Tudo modos de encriptação que o AES pode usar. Com esta informação as nossas suspeitas acerca do processo main ser o autor das encriptações feitas no computador da Sally aumentaram.

De seguida fomos à procura de qualquer indício no disco da Sally deste programa "main". Similarmente à questão anterior de encontrar a fonte, qual e como o malware afetou o computador em questão, começámos por analisar o conteúdo do diretório "/home/sally/". Começámos inicialmente por encontrar informações a navegar nos diretórios utilizando, respetivamente, dois comandos do TSK: "mmls" e "fls" (flags "-Fra"). Pudemos verificar naturalmente, ao analisar a pasta "/home/sally/Documents/", que todos os ficheiros da Sally se encontravam cifrados utilizando o programa "icat", que nos devolveu a mensagem "Jason's back!" nos ficheiros cujo nome parecia "normal" e, nos ficheiros cujo nome era seguido da extensão ".encrypted", devolvendo-nos "lixo", o que indiciava que os ficheiros originais tinham sido mesmo cifrados, e que o resultado disso seriam então estes ficheiros. De seguida, focámos a nossa atenção no Desktop e encontrámos um ficheiro apagado ("realloc") que recuperámos através do "icat" de um vídeo de análise de células cancerígenas. Inicialmente achámos que o malware pudesse ter apagado este vídeo; no entanto, chegámos à conclusão de que a própria Sally o pudesse simplesmente ter apagado por não lhe interessar mais. Cremos, portanto, que o "scope" do ransomware não foi para além da pasta "/home/sally/Documents/". Adicionalmente, como foi mencionado anteriormente, encontrámos também informação relativa ao perfil da Sally tanto do Thunderbird como do Firefox, de entre outras aplicações no diretório "/home/sally/.cache". Finalmente, após analisar outros diretórios sem qualquer

informação útil, encontrámos na pasta "/home/sally/Downloads/" um ficheiro binário com o nome de "main" que tinha sido apagado ("realloc"). Tendo encontrado informação bastante relevante partimos para a criação de um script em python chamado ""parse_disk.py" que recuperava toda a informação do "/home/sally/", pois além de ter sido aqui o local do crime (encriptação) era também este diretório que continha os ficheiros mais importantes para a Sally.

Tendo corrido o script "parse_disk.py" que fez "icat" de todos os ficheiros na diretoria mencionada acima, não conseguindo abrir qualquer um dos ficheiros encriptados na pasta "/home/sally/Documents/", e também não conseguindo concluir nada de novo do binário "main" encontrado na pasta "/home/sally/Downloads/", partimos para a análise do directório "/home/sally/.cache". Neste diretório decidimos primeiro restaurar o perfil da Sally no Firefox. Aqui não encontrámos nada de relevante para o caso, apenas um histórico de navegação normal de acordo com a investigação levada a cabo pela Sally. No entanto, ao recuperar o seu perfil do Thunderbird é que obtivemos resultados mais interessantes. Aqui, como mencionámos na questão anterior, encontrava-se um e-mail com um executável também chamado "main" fazendo-se passar por uma atualização de segurança. Suspeitamos que este executável tenha sido o mesmo executável que foi apagado (após a sua execução, cremos nós) encontrado na pasta "/home/sally/Downloads/". O nome do remetente que usou o serviço de e-mail "Proton Mail": "jason_halloween" correspondia também às referências feitas na mensagem obtida ao abrir qualquer um dos ficheiros. Tendo obtido estas informações ficou cada vez mais evidente que foi este executável o responsável pela encriptação dos ficheiros no computador da Sally. É de mencionar que os ficheiros encriptados que obtivemos encontram-se na pasta "evidence_artifacts/encrypted_sally_documents/" e que o perfil Thunderbird se encontra na pasta "evidence_artifacts/thunderbird_mail_evidence".

Tendo obtido os ficheiros da pasta Documents da Sally, partimos então para a criação de um programa de desencriptação dos ficheiros obtidos. No entanto, para decifrar os ficheiros necessitamos primeiro da chave com que estes foram cifrados, caso contrário teria que se recorrer a bruteforce. Como vimos anteriormente, na lista de bibliotecas que a função "main" importava, foram importadas bibliotecas de encriptação com o algoritmo "AES". Sendo um algoritmo de cifra simétrica basta-nos encontrar a chave com que foi encriptado para decifrar os ficheiros. Dado que esta chave foi usada pelo processo para encontrar os ficheiros ela deve estar algures escrita no espaço de memória da Sally. Com isto em mente, procuáamos informação na Internet que nos ajudasse a realizar este processo. Encontramos o URL do seguinte git: "https://github.com/makomk/aeskeyfind.git", o qual clonámos para o nosso sistema. Este repositório codifica um programa chamado "AESKEYFIND" que, como o próprio nome indica, encontra-nos uma chave AES após lhe darmos como input o espaço de memória pretendido, que foi o que fizemos. Ao aplicar este programa ao espaço de memória da Sally encontramos diversas chaves "AES", mas só uma chave era apenas de 128 bits: 47683b9a9663c065353437b35c5d8519. Esta era a única chave de 128 bits (32 caracteres), dado que as outras encontradas tinham todas 256 bits (64 caracteres), tal como se pode observar no ficheiro "evidence_artifacts/aeskeyfind/aeskeyfind.png". Dado que este programa apenas funciona em distribuições Ubuntu, utilizámos o Ubuntu 16.04 para obter a chave. Uma exceção clara ao Software Kali que foi utilizado em todos os outros procedimentos feitos então até agora.

Tendo sido a chave obtida, decidimos desenvolver um script chamado de "decrypt_files.py" que basicamente agarra em cada ficheiro cifrado encontrado e decifra-os utilizando a chave AES de 128 bits encontrada utilizando cifra por blocos em "Counter Mode" (CTR). A cifra por blocos em CTR mode foi, novamente, uma das bibliotecas importadas pelo processo "main" para a pasta "/tmp/_MEIlXS6RU/" indicada acima. Ao correr o script "decrypt_files.py" no diretório "/home/sally/Documents/" todos os ficheiros foram decifrados para a diretoria escolhida no input. Este ficheiro "decrypt_files.py" encontra-se no directório "auxiliary_items/file_recovery_aux".

Já depois de ter obtido os ficheiros originais tentamos também usar "carving" tools do TSK em vários dos fragmentos de informação obtidas no disco, mas sem nenhum resultado que demonstrasse qualquer nível de sucesso.

Tendo obtido os ficheiros desencriptados originais demos como encerrada a nossa procura e devolvemos todos eles à Sally. Todos os ficheiros que foram decifrados encontram-se no diretório "evidence_artifacts/", sendo que os cifrados correspondentes (o que retirámos do disco da Sally originalmente) também lá estão.

3 What can you tell about the identity of the attacker?

Após termos decifrado todos os ficheiros da Sally alvejados pelo ransomware decidimos ir em busca da identidade do atacante.

Os nossos pontos de partida foram qualquer resto de informação escondida no disco, no espaço de memória ou no e-mail que foi enviado à Sally. Após procurar detalhadamente várias dos processos encontrados em memória com as ferramentas das no volatily e ter novamente procurado por qualquer rasto de ficheiros que pudessem dar à identidade do atacante no disco sem sucesso, decidimos focar a nossa atenção no aparentemente único contacto direto que houve entre a Sally e o criminoso, o e-mail que lhe foi enviado. Esta mensagem revela alguns rastos de uma possível identidade do atacante porque quem o mandou assinou como "Sid Wilkes". “Sid Wilkes” pode ser apenas um nome inventado, mas, no entanto, pode realmente ser o nome de um funcionário pertencente ao suporte técnico do departamento de tecnologias de informação. Suspeitamos, portanto, que caso “Sid Wilkes” exista, então o criminoso sabe da existência deste, o que quer dizer provavelmente que o criminoso é alguém familiar à universidade, dado o seu conhecimento de um departamento interno do campus e do nome de um dos seus funcionários. Suspeitamos, portanto, como já deveria ser implícito, que esta a assinatura tenha sido forjada, ou seja, que o verdadeiro remetente não tenha sido a pessoa que consta na assinatura do e-mail (“Sid Wilkes”). Este e-mail, de facto, não aparenta ser da mesma pessoa que alegadamente assinou o e-mail, uma vez que o e-mail é "jason_halloween@protonmail.com". Assim, por inferência, podemos deduzir que o endereço de e-mail anterior não seria considerado um e-mail apropriado para um funcionário da assistência técnica do campus de bioquímica da universidade. Além do mais, só pelo facto do endereço pertencer a um domínio do protonmail podemos inferir que este endereço não pertence à universidade, pois devido à dimensão das universidades hoje em dia e às arquiteturas de sistemas e internet em voga, os funcionários da universidade usariam um e-mail com o domínio dos servidores de e-mail da mesma. Finalmente, dado este e-mail conter como anexo o programa que aparentemente cifrou os documentos da Sally podemos inferir que este provavelmente pertence ao atacante. O e-mail em causa encontra-se num “screenshot” presente no ficheiro "thunderbird.png" no diretório “/evidence_artifacts/thunderbird_mail_evidence/”. Entretanto, pesquisámos sobre como “trackear” IPs no thunderbird de forma a descobrir a origem dos e-mails, e encontrámos num fórum que a macro "CTRL+U" nos mostra o source code da mensagem. Após fazermos isso, gravámos esse resultado no ficheiro "thunderbird_mail2ip.txt" e pesquisámos aplicando o seguinte regex "[0-9]*[.][0-9]*[.][0-9]*[.][0-9]*", que nos fez highlight no sublime de IPs. Após isto, começámos a analisar o fluxo do mail pelo último "Received", até ao primeiro, o que nos dá o caminho que o e-mail tomou desde a sua origem até ao seu destino. Concluímos que o nó de entrada do e-mail é o nó cujo IP é 185.70.40.136. Após isto, pingámos o servidor, verificando que este é acessível. Depois, corremos o comando "nslookup 185.70.40.136" e determinámos que o seu URI é: "mail-40136.protonmail.ch.". Pode-se verificar isto na imagem "protonmail_nslookup.png". Um dos passos que poderíamos tomar mais tarde seria pedir um mandato para este servidor específico da protonmail, analisar os logs do servidor e tentar encontrar provas que nos indiquem a origem daquele e-mail.

4 Elaborate a timeline of the most significant events of the case.

Hora do mail (retirado do Thunderbird)	12-11-2018, 16:53 2018-11-12 16:53:--.---------
Hora do download	Entre a hora do mail recebido e a hora em que o programa "main" foi corrido (indeterminado)
Hora do processo main a correr (de acordo com o Linux_pslist, ambos os processos main encontrados começaram a correr ao mesmo tempo)	2018-11-12 17:15:45
Hora da encriptação dos ficheiros (retirado com o comando istat nos ficheiros ".encrypted" (e os ficheiros que dizem "Jason's back") da Sally, do primeiro ao último ficheiro passa muito pouco tempo, o que indicia automatização, no entanto ter em atenção o que referimos na nota a seguir)	(do primeiro ao último, todos têm os últimos 3 dígitos sendo 0s) 2018-11-12 17:20:14.285643000 – 2018-11-12 17:20:15.261646000
Hora em que o processo main apagou o ficheiro "/home/sally/Downloads/main"	2018-11-12 17:24:38.094254576

Nota: podemos concluir a partir dos tempos perfeitos (isto é, tempos que acabam com 0s) que o malfeitor tentou esconder ou forjar estes tempos. A única razão para a qual vemos que isto seria vantajoso ao malfeitor seria esconder os tempos reais de acesso aos ficheiros na sua escrita.

Assinado por:

Inês Barral Paiva, nº 77926

João Pedro Meira, nº 82014

José Eduardo Brás, nº 82069

No dia: 25-11-2018

Apêndice:

Dentro da pasta "auxiliary_items", temos uma pasta chamada "file_recovery_aux" nos quais se encontram os scripts auxiliares descritos acima, e também o script "parse_disk.py", que foi utilizado para retirar os ficheiros pretendidos do disco da sally, incluindo ficheiros já apagados (basta apenas dar o output da ferramenta "fls" ao parser e este vai buscar o ficheiro correspondente). Com esta última tool apanhámos os ficheiros correspondentes à directoria "home/sally/" e reconstruímo-los no nosso sistema. Adicionalmente, a pasta "auxiliary_items" contém também outra pasta, chamada "volatility_output", no qual temos os outputs do comando volatility que foram utilizados e se revelaram úteis durante a experiência. Finalmente, um ficheiro de relatório mais informal chamado "logs.txt" encontra-se nesta mesma pasta ("auxiliary_items").

Os ficheiros originais e os resultados dos scripts mencionados acima encontram-se na pasta "evidence_artifacts" (em "encrypted_sally_documents/" e "decrypted_sally_documents/"), excepto os do script "parse_disk.py", porque o resultado era demasiado grande para a submissão. No entanto, os conteúdos das directorias ".cache" e "home/sally/" constitui prova e, como tal, deveria estar na pasta "evidence_artifacts". Para além disto, dentro desta pasta temos outra pasta chamda "thunderbird_mail_evidence", na qual apresentamos o e-mail suspeito encontrado, o programa main e o "source code" do protocolo de mail (obtido através de "CTRL-U" no thunderbird) que nos dá informação acerca do servidor de e-mail que atendeu o pedido de jason_halloween e que poderá, eventualmente, ter logs relevantes para descobrir a verdadeira identidade do criminoso. Adicionalmente, dentro da mesma pasta, está também uma pasta "aeskeyfind/" com um ficheiro "aeskeyfind.png", o qual contém o output do programa "aeskeyfind" quando dado como input a memória do computador da Sally.

Nota: apesar de os ficheiros da pasta "volatility_output" constituírem provas, decidimos pôr nos itens auxiliares pois nos serviram de base para muitas das descobertas. Não deixam, por isso, de ser provas.