INSTITUTO SUPERIOR TÉCNICO
Departamento de Engenharia Informática
Ciber Segurança Forense
MEIC / METI 2018-2019 - 1º Semestre

Digital Forensics Report

Autores: (Grupo 6) Ruben Condesso (81969); Miguel Carreiro (82012)

GROUP I – Network trace analysis

1

1.1 Indicate the complete URL of the original web request that led to the client being compromised

O URL completo é o seguinte: *http://10.20.0.111:8080/banking.htm*.

1.2 What file type was requested in the final web request to the malicious server?

Foi um ficheiro do tipo GIF. Ao abrir a captura *lab3-pcap1.pcap*, usando o *Wireshark*, verificámos que foram feitos imensos pedidos a diversos *websites*, sendo normal ter havido uma grande atividade, dado o contexto do enunciado. Decidimos averiguar os pedidos feitos, na rede local, e dessa forma, usámos o filtro *ip.src=10.20.0.0/24 && ip.dst=10.20.0.0/24 e http*.

Posto isto, apurámos que houve um *request URI* para o *banking.htm*, do PC (10.20.0.165) para o possível servidor malicioso (10.20.0.111). A resposta deste último, fez com que tenha havido um redireccionamento do *site* pedido. Tal, fez-nos suspeitar que esta ação pode estar diretamente ligada a um possível envio de um ficheiro malicioso (por parte do servidor), sendo que desta forma, o atacante pode ter "obrigado" o PC a visitar um *site* de forma intencional, e assim poder ser infetado. De seguida, é feito um *request* de um ficheiro do tipo *.gif*, de onde o servidor responde afirmamente esse pedido, e desta forma, concluímos que o tipo de ficheiro pedido no *final web request*, para o servidor malicioso, foi um GIF.

Estas interações estão ilustradas na Figura 1 - *captura_http_malicioso* (/*Screenshots*).

1.3 What is the number of the first frame that indicates that the client has been compromised?

O número da primeira *frame* é 4665, que pertence à resposta do servidor malicioso referente ao pedido do PC, depois de ter havido o redireccionamento referido na resposta anterior. O cliente pede a página maligna (involuntariamente), o servidor malicioso devolve lhe o conteúdo da página, e por sua vez, o cliente recebe essa resposta e retorna *ok*. Considerámos esta última *frame* (pacote), como aquela que iniciou o ataque propriamente dito.

1.4 At one point, the malicious server sends a malicious file to the client. What type of file it is?

O servidor malicioso enviou um ficheiro malicioso, do tipo *Windows executable* (letra a). Para chegarmos a essa conclusão, começamos por filtra os pacotes cuja a origem pertencem ao servidor (10.20.0.111). De seguida, analisámos os pacotes do tipo TCP, que continham a *flag* PSH, sendo que esta corresponde ao início da transferência de ficheiros.

Selecionámos a primeira (por exemplo), e seguimos o *TCP Stream*. Extraímos o conteúdo do *TCP Stream*, no formato *RAW*, e guardámos num ficheiro (*buffer_tcp*). Posto isto, usámos o *foremost* para tentar encontrar as extensões *.exe*, *.js*, *.pdf*, *.docx*, *.doc* e *.gif*, e foi encontrado um ficheiro (*00000000.dll*), ou seja, do tipo executável.

Usámos o *site virustotal.com*, para averiguar se esse ficheiro contém *malware*. Foi, de facto, encontrado um *cavalo de troia*. O *output* desse *site* encontra-se ilustrado na *Figura 2 - Análise_trojan* (*/Screenshots*).

1.5 What is the SHA1 hash of the malicious file?

Para encontrar a *SHA1 hash*, executámos o comando *sha1sum 00000000.dll*, cujo o resultado foi: 94adf100411a80076192766a214e0ff92da13ab7.

1.6 What vulnerable software has been exploited (in the following format, Firefox 3.5, Firefox 3.6, Word 2010, IE7, Safari2, Chrome2, AdobeReader, IE9)?

No seguimento do filtro usado na pergunta 1.2, abrimos um dos pacotes, cuja a origem fosse do PC (10.20.0.165), sendo que estes continham informação sobre o *browser* usado. No parâmetro *User-Agent*, estava descrita a seguinte informação: *Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)*. Usámos o *site* developers.whatistmybrowser.com, para descobrir qual o *browser* usado pelo PC. Obtivemos como resposta que essa descrição pertence ao *Internet Explorer 6*, e foi acedido no sistema operativo *Windows XP SP2*.

A Figura 3 - *software_usado* (*/Screenshots*) ilustra estes resultados.

1.7 When the capture ends, is the client still connected to the malicious attacker?

Uma ligação TCP é iniciada com o "*Three way handshake*" e é terminada pela sequência FIN, FIN-ACK ou RST. Podemos verificar que na captura em causa, os últimos pacotes transmitido entre o PC e servidor malicioso, usando TCP, são do tipo ACK e PSH. Logo, não foi retornado nenhum pacote do tipo FIN, FIN-ACK ou RST, e desta forma, podemos concluir que o PC ainda estava conectado ao servidor malicioso, quando a captura terminou.

A Figura 4 - *fim_captura* (*/Screenshots*) remete a esta conclusão.

1.8 Can you indicate the corresponding CVE security bulletin that covers the vulnerability that was exploited here (answer in form of CVE-$year-$number)?

O CVE correspondente à vulnerabilidade explorada, neste contexto, foi o *CVE-2010-0249*. O *CVE-2011-3887* é referente ao *Google Chrome*, enquanto o *CVE-2010-3178* é referente ao *Mozilla Firefox*, *Thunderbird* e *Seamonkey*, logo podíamos excluir estas duas hipóteses. Tanto o *CVE-2010-0249* como o *CVE-2010*-1127, são referentes ao *Internet Explorer 6*, no entanto, apenas o primeiro atua no sistema operativo *Windows XP SP2*. Além disso, lendo a descrição de ambos os CVE's, verificamos que o *CVE-2010*-1127 tem como consequência da vulnerabilidade *disruption of service* (*Denial of Service*), o que não é o caso para o nosso contexto, enquanto, o *CVE-2010-0249* é relativo à seguinte especificação: *Victim must voluntarily interact with attack mechanism; Allows unauthorized modification*, *disclosure of information and/or disruption of service.* Logo, podemos concluir que estas características se encaixam no nosso contexto. Estas informações foram retiradas do *site* cve.mitre.org.

1.9 From the capture, it is clear that the attacker gets a certain form of access (i.e. the interface), what (type of) access does the attacker "get" on the client?

Já tínhamos descoberto, anteriormente, que o servidor malicioso tinha usado um Cavalo de Troia, de forma a infetar o PC. Posto isto, começamos por verificar os portos usados na captura, sendo que o porto utilizado que nos chamou à atenção, foi o 4444.

De seguida, voltamos ao *output* dado pelo *site* virustotal.com, onde era dado (por vários *antivírus*) a descrição (e nome) do vírus em causa. Utilizámos a informação dada pelo AVG (*Multi:Swrort-A [trj]*) relativo ao vírus, e pesquizando no *google*, chegámos à conclusão que este *trojan* esconde-se em programa, e quando instalado, tem a capacidade de instalar programas maliciosos no PC. De notar que usámos a informação dada pelo AVG por ser dos *antivírus* mais utilizados atualmente, e logo mais fiáveis.

2

2.1 What is the frame that indicates that something strange might be going on?

A *frame* que indicia que algo de estranho está a acontecer é a *frame* 8 (o primeiro *ping*). Não é esta *frame* isolada que poderá constituir o início de uma ameaça, mas sim por ser a primeira de um conjunto de *frames* com o mesmo propósito: fazer *pings* a todos os endereços da rede, de forma a descobrir quais é que estão a ser usados. Desta forma, este pode ser o início de um *scan* dos *hosts* da rede, o que pressupõe que alguém poderá estar interessado em tirar partido dessa vulnerabilidade da mesma.

Estes resultados encontram-se ilustrados na Figura 5 - *pings_estranhos* (/*Screenshots*).

2.2 What tool is generating this traffic?

A ferramenta usada para gerar o tráfego em causa, foi o *nmap*. Esta é uma ferramenta *open-source*, que tem como objetivo fazer auditorias a uma rede, podendo vir a descobrir falhas e vulnerabilidades na mesma. Este tráfego é referente ao envio de mensagens ICMP, do tipo 8, para os vários endereços IPs, e tais mensagens podem ser geradas pelo *nmap* (ICMP *ping types -PE, -PP ou -PM*), que estará à espera de *echo reply* (tipo 0), dos *hosts* disponíveis. Esta prática é bastante adequado para ser usada em redes locais, que por sua vez, coincide com o nosso contexto. A Figura 6 - *ping_tipo8* e a Figura 7 - *ping_tipo0* (/*Screenshots*) remetem à ilustração do tipo de mensagens trocadas, referidas anteriormente.

2.3 What does this frame constitute the beginning of?

A *frame* em causa constitui o início de um TCP *scan*. No seguimento do raciocínio da resposta à pergunta 2.1, depois de ter sido feito os vários *pings* aos *hosts* da rede, houve uns que responderam em como estavam ativos e outros que não. Podemos verificar que depois é feito um TCP *scan* aos que responderam afirmamente ao *ping* (usando o filtro *icmp.type ==0 && (icmp.code==2*), devido às vários pacotes TCP trocados entre PC e servidor depois dessa resposta. Filtrando a captura para um *host* ativo (por exemplo, o 192.168.10.11), verificamos as mensagens do tipo SYN, ACK e RST entre os dois, que remete TCP *connect scan* (-sT). Posteriormente, também podemos ver que há um UDP *scan*, no entanto, como a pergunta refere-se ao que a *frame* referida na resposta à 2.1 constituiu, concluímos que foi, de facto, um TCP *scan*, sendo que este aconteceu no seguimento dos *pings* feitos logo no início da captura (imediatamente a seguir a ter-se averiguado quais os *hosts* ativos).

Este *scan* encontra-se ilustrado na Figura 8 - *tcp_scan* (/*Screenshots*).

2.4 A switch was removed from the command to improve the speed, what was this switch (just the letters, case-sensitive)?

Segundo a captura *wireshark*, podemos verificar que o *scan* TCP começou por volta dos 255.5 (n=2085) segundos, ou seja, nos 4 minutos e 26 segundos que vai ao encontro do primeiro intervalo de tempo referenciado no enunciado. O outro *scan* TCP começou por volta dos 674.6 (n=17550) segundos, ou seja, nos 11 minutos e 14 segundos.

Os *switches* utilizados no *nmap* foram o -sS (*Scan TCP SYN*) e o -sU (*Scan UDP*). O *Scan TCP SYN* permite analisar milhares de portas TCP que não estejam restritas por *firewalls*, e não é obstrutiva uma vez que nunca completa a ligação TCP. Neste *scan*, é enviado um pacote SYN, da mesma forma que uma ligação real é aberta, e depois aguarda-se pela resposta. Caso seja recebido um pacote *SYN/ACK*, ou se for recebido um pacote *SYN* (sem a *flag ACK*), então considera-se que a porta está aberta, enquanto que, caso seja recebido um pacote RST, a porta é considerada como não-ouvinte (*non-listener*). Se não for recebida nenhuma resposta, ou se for recebido um erro *ICMP unreachable* (tipo 3, códigos 0,1,2,3,9,10 ou 13), a porta é considerada como filtrada. Na captura é possível observar o envio de vários pacotes *SYN* para várias portas, sendo que as respostas recebidas também correspondem com este comando.

Na captura foi também possível detetar o envio massivo de pacotes *UDP* para várias portas, esperando uma resposta por parte dos *hosts*. Segundo a documentação disponível no *site* do *nmap*, o comando -sU envia um cabeçalho *UDP* vazio para todas as portas que se pretendem testar, aguardando depois uma resposta. Caso seja recebido um erro *ICMP Port Unreachable* (tipo 3, código 3), considera-se a porta fechada, caso sejam recebidos outros erros *unreachable* (tipo 3, códigos 0,1,2,9,10 ou 13), considera-se a porta filtrada, caso seja recebida um pacote *UDP*, considera-se a porta aberta, e, por último, caso não seja recebida nenhuma resposta, a porta é considerada aberta, ou eventualmente existe algum filtro a bloquear a ligação. A captura possui vários cabeçalhos *UDP* vazios que foram enviadas para várias portas diferentes, bem como as respostas dos *hosts*, o que atesta a utilização deste *switch* no *nmap*.

O *nmap* permite que o *scan UDP* seja executado em conjunto com outros tipos de *scan TCP*, logo é possível afirmar que os dois *switches* podem ser utilizados em conjunto. Após o primeiro *scan*, o *switch -sU* (*Scan UDP*) foi removido, ficando apenas o *switch -sS*, pois observando as capturas respeitantes ao *scan* final, é possível verificar que apenas são enviados pacotes *TCP*, sendo que não são enviados pacotes *UDP*, confirmando assim que no segundo *scan* apenas foi executado o *switch -sS*.

2.5 What switch was added to the final scan (case-sensitive)?

O *switch* utilizado foi o -b (FTP *bounce scan*). Este *switch* utiliza uma vulnerabilidade de uma funcionalidade do protocolo FTP, o *proxy* FTP, que permite ligar a um servidor FTP e solicitar o envio de ficheiros para um terceiro servidor. Esta vulnerabilidade permite, entre outras coisas, efetuar um scan das portas dos outros *hosts*.

Este *switch nmap* funciona enviando um arquivo para cada porta a testar, analisando depois a resposta do servidor, sendo que, para que este *switch* funcione, é necessário fornecer o endereço do servidor FTP, e, caso se pretenda, o *username*, *password* e a porta (caso não sejam fornecidos estes dados, o *username* será *anonymous*, a *password* -wwwuser@ e a porta 21).

Na captura encontra-se um pacote FTP (nº 47347 e 47348), que envia um *request* em que o *username* é *anonymous*, sendo que o servidor solicita o envio da *password* correspondente, em que o cliente retorna a *password* IEUser@ (nº 47371 e 47372). Embora o *username* seja igual ao pré-definido pelo *nmap*, a *password* é diferente, o que poderá significar que o *nmap* recebeu não só o endereço do servidor FTP, mas também o *username* e *password*. Em suma, estes dados permitem afirmar que, no final do último *scan*, foi utilizado o *switch -b*, que corresponde ao FTP *bounce scan*.

GROUP II – Website fingerprinting over Tor

1 Indicate: (a) the selected feature set, and (b) the accuracy of your classifier using this feature set. Give an account of other features you have tried before converging into this feature set. Explain how you've modified the source code in order to compute your proposed feature set.

O nosso primeiro passou foi encontrar todas as possíveis *features* que se enquadrassem para o nosso contexto. Desta forma, considerámos 4 *features*: *Transmissions size features*, *Transpositions*, *Packet Distributions* e *Bursts*. Colocámos estas *features* na função *extract*, do código *fextractor.py*, onde cada uma está devidamente identificada com comentários.

De seguida, considerámos diversas combinações das *features*, de forma, a chegar ao melhor *feature set* possível. Tais combinações feitas, estão ilustradas na tabela 1. Podemos concluir que a melhor *feature set* encontrada, é a combinação entre os *transmissions size* e *transpositions*, que por sua vez, origina uma precisão de 0.8194, que por sinal é superior a qualquer uma calculada.

Para K=5							
Feature set	1	2	3	4	5	6	7
Transmissions size feature	**X**	**X**	**X**	**X**			
Transpositions	**X**	**X**	**X**		**X**		
Packet Distributions	**X**	**X**				**X**	
Bursts	**X**						**X**
Accuracy	0,8105	0,8083	**0,8194**	0,4	0,7383	0,6089	0,5861

Tabela 1 - Valores da precisão do classificador, segundo as várias *features* utilizadas, com K=5

2

2.1 Can you spot a trend in the accuracy of the classifier as k increases? Explain the variations observed.

k	1	2	5	10	15
ACC	0.9211	0.8866	0.8161	0.7411	0.6738

Tabela 2 - Valores da precisão do classificador, usando o *feature set* escolhida no exercício anterior, para os vários valores de K

Como está ilustrado na tabela 2, podemos verificar que, para o *feature set* escolhido no exercício anterior, à medida que aumentámos o valor do K, há uma diminuição gradual do valor da precisão do classificador em causa. Desta forma, podemos ver que o valor da precisão é mais elevado para K=1 (0.9211), o que pode ser facilmente explicado pelo facto do classificador ter que correr uma menor distância no seu processo de classificação, comparativamente com os restantes valores de k, o que faz com o seu processo seja muito menos complexo, e daí obter um valor mais elevado para a precisão. No entanto, isto não quer dizer obrigatoriamente que este valor de K é o melhor, de todos os considerados, pois sendo o que apresenta o valor mais baixo (o que percorre a menor distância), é também o que é mais susceptível aos erros do ruído.

Então tem de haver este balanceamento na hora de escolher o meu valor de K a utilizar, e assim, o valor da precisão para K=2 (0.8866) será certamente um valor a considerar. Por um lado, é menos susceptível a erros, comparativamente a K=1, mas apresenta um maior grau de complexidade (e logo menos precisão). Podemos concluir que há um grande equilíbrio entre estes valores de K. Os restantes valores de K não deverão ser considerados devido aos baixos níveis de precisão que apresentam e à elevada complexidade de implementação que têm.

Os resultados obtidos na tabela 2 dizem respeito às figuras 9, 10, 11, 12 e 12 (/*Screenshots*).

2.2 Can you spot any trend in the TPR/FPR trade-off alongside the variation of k and nm? Justify.

Nm%\ K		1	2	5	10	15
0.1	*TPR*	0.9899	0.8404	0.7061	0.6197	0.5534
	FPR	0.1513	0.0364	0.0050	0.0033	0.0028
	Accuracy	0.9111	0.9033	0.8511	0.8086	0.7758
	TPR/FPR	6.5426	23.0879	141.22	187.7879	197.6429
0.2	*TPR*	0.9833	0.8100	0.6953	0.5974	0.5159
	FPR	0.1520	0.0299	0.0055	0.0028	0.0022
	Accuracy	0.9067	0.8919	0.8456	0.7978	0.7572
	TPR/FPR	6.4691	27.0903	126.4182	213.3571	234.5
0.4	*TPR*	0.9812	0.7952	0.6795	0.5768	0.4944
	FPR	0.1457	0.0271	0.0061	0.0028	0.0011
	Accuracy	0.9089	0.8861	0.8375	0.7875	0.7469
	TPR/FPR	6.7344	29.3432	111.3934	206	449.4545
0.6	*TPR*	0.9783	0.8115	0.6881	0.5838	0.4999
	FPR	0.1314	0.0277	0.0066	0.0032	0.0017
	Accuracy	0.9164	0.8939	0.8417	0.7908	0.7492
	TPR/FPR	8.0222	29.2960	104.2576	182.4375	294.0588
0.8	*TPR*	0.9747	0.8117	0.6821	0.5796	0.4894
	FPR	0.1215	0.0292	0.0061	0.0028	0.0017
	Accuracy	0.9208	0.8936	0.8389	0.7889	0.7442
	TPR/FPR	8.0222	27.7979	111.8197	207	287.8824

Tabela 2 - Valores do TPR, FPR, TPR/FPR e *accuracy*, para os vários valores de K e de treino

Com o auxílio da tabela 2, podemos verificar uma tendência de descida dos valores de TPR e FPR á medida que os valores de K aumentam. Também se verificou uma descida da *accuracy* á medida que os valores de K aumentam.

O aumento dos valores de nm levaram a uma ligeira diminuição dos valores de TPR e FPR. Em relação aos valores da *accuracy*, apurámos que o seu valor desce até um determinado valor de nm, e seguidamente, sobe com o aumento da percentagem de treino. Essa inversão de padrão ocorre em valores diferentes de nm para os vários valores de K. Por exemplo, no caso em que K=1, o valor da *accuracy* desce até nm=0.2, e a partir daí, sobe até nm=0.8, e no caso em que K=5, o valor da *accuracy* desce até nm=0.4, voltando a subir até nm=0.8.

A altura a que esta inversão ocorre, deve-se ao facto de ir havendo um equilíbrio entre o a diminuição dos *true positives* e a diminuição dos *false negatives*. Ou seja, estamos a diminuir o primeiro parâmetro (o que é mau), e diminuir o segundo (o que é bom). Logo, a altura dessa variação dita o melhor equilíbrio calculado entre esses dois parâmetros, que depois reflete-se na *accuracy*.

2.3 According to your results in Table 2, which configuration leads to a more successful attack? Do you think an attack with such a TPR / FPR trade-off may be effective in practice? Justify.

Idealmente, um ataque bem sucedido deve apontar a um TPR grande, de forma a garantir o maior número de *sites* monitorizados que foram identificados, e em simultâneo ter um FPR baixo, de forma, a minimizar os riscos de classificar acessos a páginas não monitorizadas, bem como a páginas monitorizadas. Consultado a tabela 2, verificámos que a configuração que apresenta melhor resultados é aquela cujo o resultado de TPR/FPR é maior, que por sua vez, é 449.4545, com nm=0.4 e k=15. Temos então um caso, onde o classificador tem de percorrer uma distância enorme e o seu treino é inferior a 0.5, e assim temos a melhor configuração possível para o ataque.