INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Forensics Cyber Security

MEIC / METI 2018-2019 – 1st Semester

Digital Forensics Report

Daniel Chaves - 81651

Afonso Santos - 81861

Manuel Sousa - 84740

GROUP I – Network trace analysis

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

The original URL is http://10.20.0.111:8080/

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

The requested type was for a GIF (e.) file, but the answer was a redirect to the exploit page.

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

The frame number is: 4722

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

The malicious server sends a Windows executable (a.).

1.5 What is the SHA1 hash of the malicious file?

The sha1 of the malicious file is: 94adf100411a80076192766a214e0ff92da13ab7

In order to get this file we first analysed an html file that was the origin of this intrusion. The file banking.htm (IID A-01-BANKING-HTM) contained another JavaScript file (IID[1] A-02-VIRUS-JS) that represented the actual exploit. After the exploit ran, a TCP stream was intercepted containing a DLL file (IID A-03-VIRUS-DLL). This TCP stream was retrieved from wireshark, and then we used Foremost to get the malicious DLL file from this stream.

This DLL file is highly marked as malware by VirusTotal that strengthens our thesis about something malicious was intercepted in this communications.

[1] IID is an abbreviation of Investigation Identifier

Investigation ID	A-01-BANKING-HTM
File name	banking.htm
File Type	HTML document text, ASCII text, with very long lines
File Size	12612 Bytes
sha1 sum	e184ac3ff41b2da046e1bcc546e9df7939c997bd
md5 sum	a576776febc36992086fe9a7d2662836

Investigation ID	A-02-VIRUS-JS
File name	virus2.js
File Type	HTML document text, ASCII text, with very long lines
File Size	4045 Bytes
sha1 sum	cea842cea2712c48af16660fa87862563878de12
md5 sum	88e1d1e8a4810cb5c672367e96fca6fa

Investigation ID	A-03-VIRUS-DLL
File name	virus.dll
File Type	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
File Size	752128 Bytes
sha1 sum	94adf100411a80076192766a214e0ff92da13ab7
md5 sum	216c9d064601b4d2066f4573ac20d656

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)?

The vulnerable software that has been exploited was IE6 as described in the CVE (question 1.8).

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

Yes because at the end this interception the client is still communicating with the malicious attacker and also we didn't saw a TCP fin packet being intercepted (for this tcp channel).

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)?

The CVE is : CVE-2010-0249 , because the OS and browser version match, and we can see that the deobfuscated payload executes the same exploit as the one on https://www.exploit-db.com/exploits/11167 (with a different payload).

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?

The attacker gets arbitrary code execution through the given CVE. The payload of that Remote Code Execution (RCE) is encoded on the encoded strings on virus2.js (IID A-02-VIRUS-JS). This RCE allows the attacker to execute code as the user that is running the browser. This vulnerability allows an attacker to establish a remote connection with an infected machine by running the virus.dll executable (IID A-03-VIRUS-DLL).

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

The frame number is: 8

1.11 What tool is generating this traffic?

The tool that generated this traffic was nmap. Besides the general scanning that is going on, we also analysed some http requests where the user agent is "User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)". We also ran some of nmap features in a contained environment to compare with the intercepted traffic.

1.12 What does this frame constitute the beginning of?

This frame constitutes the beginning of a Ping Scan (d.).

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

The removed switch was "sU" (UDP scan). We analysed the two scans and concluded that UDP packets have disappeared in the second one. We also ran nmap in a contained environment without this switch to comprove both scenarios.

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

The added switch was "A" that will add OS detection, version detection, script scanning, and traceroute. As in the last step, we ran nmap in a contained environment to comprove this scenario.

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.

(a) We used 484 features:
   - (i) number of packets
   - (ii) transmission time
   - (iii) packet direction fractions
   - (iv) first 30 in and out packet bursts size distribution
   - (v) first 30 in and out packets bursts times distribution
   - (vi) first 30 in and out packet bursts size and normalized times
   - (vii) first 300 packet bursts metric from wang
   - (viii) Index of the first 100 incoming packets
   - (ix) Index of the first 100 outgoing packets

(b) 81.4%

We have modified the source code by adding code that computes this metrics in python. We changed only the file fextractor.py. It is in attach.

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	92.6	88.7	81.4	74.3	68.7

Table 1

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

nm%	1	2	5	10	15
0.1	99.7	86.3	74.5	65.4	56.2
0.2	99.5	84.6	72.5	61.3	52.4
0.4	99.4	82.8	68.6	58.9	49.3
0.6	99.2	82.4	66.9	56.5	48.1
0.8	99.0	81.8	66.5	56.0	47.3

Table 2 - TPR

nm%	1	2	5	10	15
0.1	14.8	4.4	0.89	0.44	0.28
0.2	13.4	3.9	0.77	0.33	0.22
0.4	14.0	3.6	1.05	0.44	0.17
0.6	14.1	3.8	1.04	0.44	0.22
0.8	14.3	4.1	0.94	0.28	0.17

Table 3 - FPR

nm%	1	2	5	10	15
0.1	91.7	91.1	86.8	82.5	78.0
0.2	92.0	90.5	85.9	80.5	76.1
0.4	91.8	89.8	83.9	78.8	74.6
0.6	91.7	89.5	83.0	78.0	73.9
0.8	91.5	89.1	82.9	77.9	73.6

Table 4 - 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.

The one with highest accuracy is nm=0.1, k=1. In practice, this is not the best one as it has a FPR of 14.8%. We believe that for identification purposes the one with best metric is the one that has a higher TPR/FPR ratio, as we are probably more interested in arresting criminals only if we are sure rather than incur in innocence accusations. The one with the best metric TPR/FPR is K=15,nm=0.4, so it is the one that should have the best results in practice. (We will only find through this method one out of two visitors, but it will be very unlikely to accuse someone who is not an actual visitor of being a visitor, this metric is called Positive likelihood ratio). Another possible metric for fairness in order to maximize would be:

$$TPR \times (1 - FPR) = \frac{TPR \times (FPR - FPR^2)}{FPR}$$

Because $FPR - FPR^2 \sim$ FPR, this metric does not care a lot about FPR for small FPR. We think both are good metrics, but we chose to use TPR/FPR, because it values more not misclassifying innocent people as visitors of said websites (of course the first metric we used only makes sense for a relevant TPR)

Conclusion

Regarding the group 1, we were able to find the malicious file (IID A-01-BANKING-HTM). From this file we extracted another JavaScript file containing the actual exploit (IID A-02-VIRUS-JS). This exploit explored a known vulnerability identified as CVE-2010-0249, that exploited a specific version of Internet Explorer.

In the group 2 we able to identify the tool that were generating the traffic. We made multiple tests in a contained environment in order to replicate the switches being used by this tool, and we believe to have achieved good results. With a more thorough analysis we retrieve more of nmap footprints.

Finally on part II, we believe we were able to analyze the tor network dumps and successfully identify accesses to the monitored websites, with a significant degree of confidence, showing that correlation attacks are important in forensics.

Instituto Superior Técnico, 17 November 2018

[illegible] [illegible] [illegible]

Daniel Chaves Afonso Santos Manuel Sousa