Final Report

INSE 6150: Security Evaluation Methodologies

Submitted to
MASc and PhD Mohammed Mannan

UNIVERSITÉ Concordia UNIVERSITY

GINA CODY
SCHOOL OF ENGINEERING
AND COMPUTER SCIENCE

Submitted by
Juan Carlos Borges - 26865038
Yunus Emre Aydar - 40110411

April 2021

Abstract

This paper focuses on the security evaluation of different Antivirus (AV) Products by using industry and academics techniques. As requested, below are the key findings from our process:

- Most AV products guard against man in the middle inspection and attacks.
- Some AV products secure their users at the cost of the user's privacy.
- Some AV products fail to satisfy industry standards.
- All AV products fail to notify the user that their network settings will be altered.
- All AV do offer some level of protection against ransomware attacks.

A table containing the desired characteristics is provided later in the text, along with information on how tests were executed and their individual results which lead to the characteristics table results.

Introduction

As work has shifted for most to a remote connection and the increase of security threats from online sources continue to grow, cybersecurity will only become more and more popular for all companies. One of the most popular ways to increase the security of a system is using Antivirus software; though, this paper wonders how reliable these applications really are, and do these applications really exercise what they are promoting. As such, this paper will focus on applying security validation and verification techniques onto an array of anti viruses to properly answer these concerns. The paper will follow the requested document structure and start by first providing a brief explanation of the characteristics being tested for, the methodologies and the different tools used for verifying the antivirus products. It will then display the results obtained during the processes, and what are the repercussion of said findings.

Desired UDS Characteristics

From [1], [2] and the features proclaimed by the AV manufacturers, we developed the following criteria for evaluating the effectiveness and security of the different AV products.

Usability criteria

U1 Installation-Clarity: AV products installation must clearly indicate to the user when the product is fully installed and activated. Quasi-Installation-Clarity is given to the product installation finishes but is not clear if it is active.

U2 Active-Web-Protection: The product actively protects the user web activity upon installation

U3 Clear-Notification-of-Activity: Clearly notifies the user when a file activated the antivirus. Quasi-Clear-Notification-of-Activity was given to the antivirus blocks some threads in a silent manner.

U4 Respect-Network-Configurations: AVs that do not alter the network configurations without informing the user

Security Criteria

S1 Malicious-Websites-Protection: AVs that detect and block the user from accessing the malicious/phishing websites. Malicious-Websites-Protection, was given to AVs that would detect a malicious website after the user has already opened it.

S2 Blocks-Unsafe-Downloads: AVs that detect and block unsafe files with different formats from being downloaded. Quasi-Blocks-Unsafe-Downloads, was given to AVs that only detect downloaded unsafe files.

S3 Blocks-Ransomware: AVs that detect and block most known ransomware from executing in the system. Quasi-Blocks-Ransomware is given to AVs that detect or block some of the ransomware from executing.

S4 Update-Backdoor-Resilient: AV that has methodologies in place to counter the update process backdoor exploit that is explained in the Antivirus hacker book. Quasi-Update-Backdoor-Resilient, was given to AVs that do not counter the backdoor exploit but protect the signature collection.

S5 User-Driven-Uninstall-Process-Detection: AVs that ensure through pop-up messages that the user is initiating the uninstall process, and not a malware. Quasi-User-Driven-Uninstall-Process-Detection, was given to AVs that have a single check for user driven action.

Deployability criteria

D1 *Provides-Secure-Browser*: AVs that provide a secure browser option such that TLS connections are not compromised on browsers.

D2 *Hassle-Free-Free-Service*: AVs that provide a free service without financial attachments. *Quasi-Hassle-Free-Free-Service* was given to AVs that give a free trial for a more complete service.

D3 *Closed-Web-Transactions*: AVs that maintain transactions within the AV's manufacturer domain. *Quasi-Closed-Web-Transactions* were given to AVs that shared with a single third-party domain.

Chosen test environment.

We attempted to follow the environment structure described on [2] where there would be a workstation that contains the antivirus product, a workstation to dispatch different cyber attacks to the protected workstation and a centralized collection system to record the different results form the tests as shown on Figure 1. We attempted to mimic this approach by implementing a Virtual Machine on our current workstations to hold the antivirus product. This also allowed us to ensure the same OS environment of execution on different machines.

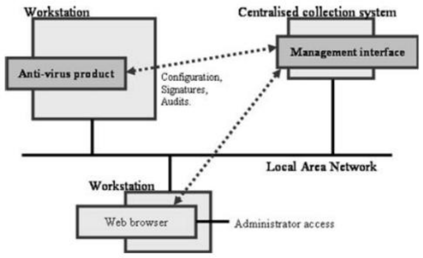


Figure 1. *Diagram indicating the ideal evaluation environment for antivirus analysis [2]*

Tools Summary

AMTSO Tests

The Anti-Malware Testing Standards Organization (AMTSO) has Security Feature Check (SFC) tools which verify that the security solution is properly configured and operating as expected that are used by professional test labs. This paper will use the following free tools provided by AMTSO, to assess the configuration and installation of the different AVs under test:

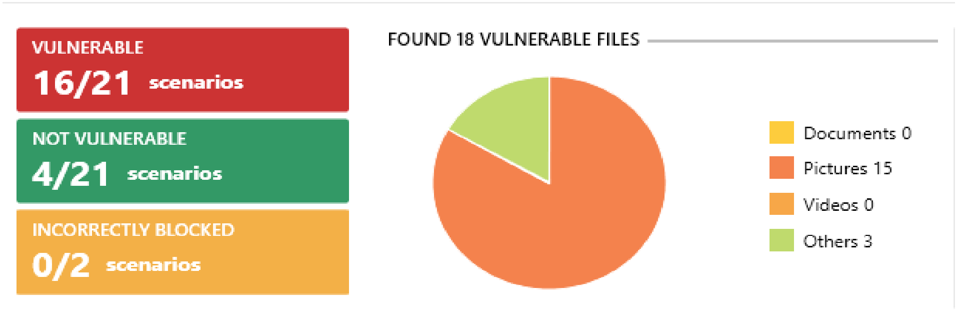
- Detect manually downloaded malware.
- Detect potentially Unwanted Applications (PUAs).
- Detect drive-by downloads of malware.
- Detect phishing pages.
- Detect compressed malware.
- Is connected to a cloud-based lookup system.

A description of these tests can be found on appendix 2.

RanSim Test

RanSim is KnowBe4`s Ransomware Simulator that helps us analyze the effectiveness of existing network protection against most common ransomware applications. We used this tool to test the antivirus programs level of ransomware protection. The tool simulates 20 ransomware infection scenarios and 1 cryptomining infection scenario, along with normal executable operations. This allowed us to observe to which ransomware attack the antivirus programs were vulnerable against as well as how many would detect false positives. The different ransomware and cryptomining operations simulated can be found in appendix 1.

After running the tool, it provides a detailed analysis on which were the failing cases by assigning it a Vulnerable, Not Vulnerable, Incorrectly Blocked, and Unexecuted tag to each simulation, an example of the results is given in Figure 2. We ensured to collect all report files in csv for the different antivirus products and combine the results into an excel table to better analyse the performance between AV products.



VULNERABLE
16/21 scenarios

NOT VULNERABLE
4/21 scenarios

INCORRECTLY BLOCKED
0/2 scenarios

FOUND 18 VULNERABLE FILES

Documents 0
Pictures 15
Videos 0
Others 3

Figure 2. Avast RanSim results

Wireshark

Wireshark is a network traffic analyzer, and an essential tool for any security professional or systems administrator. This free software lets us analyze network traffic in real time. We used the Wireshark program to test the methods under the title of The Update System mentioned in [1], as well as any suspicious traffic originating from the antivirus during a scan, installation and uninstall processes.

MITM Proxy

Mitm Proxy is a very useful tool to intercept the traffic between both client and server. The tool generally is used for debugging, testing, privacy measurements, and especially penetration. The tool is proposed by the course lecturer to improve our progress during last week of the final report. We moved on learning the tool and making use of it for our evaluation project. Technically, we downloaded certificates for the Windows operating system and edited the proxy setting manually. Then, we were able to monitor the network between client and server as man in the middle. We analyzed all antiviruses we had in our evaluation with MITM Proxy.

Testing Process

For our test process, we start by capturing the installation process of the different antivirus products using Wireshark. Once it has been successfully installed, we proceed to apply the AMTSO and RamSim tests. Using Wireshark, and repeated later with MITM proxy, we proceed to view if the antivirus applies any web calls while scanning a file on our system. Then, using Wireshark and MITM proxy again, we attempt to exploit the Update process as described in [1]. Finally, using only Wireshark, we observe to see the network traffic when the product is uninstalled, and we inspect the entire local setup to ensure there were no remnants of the product afterwards including any changes made to the network (if any).

Results

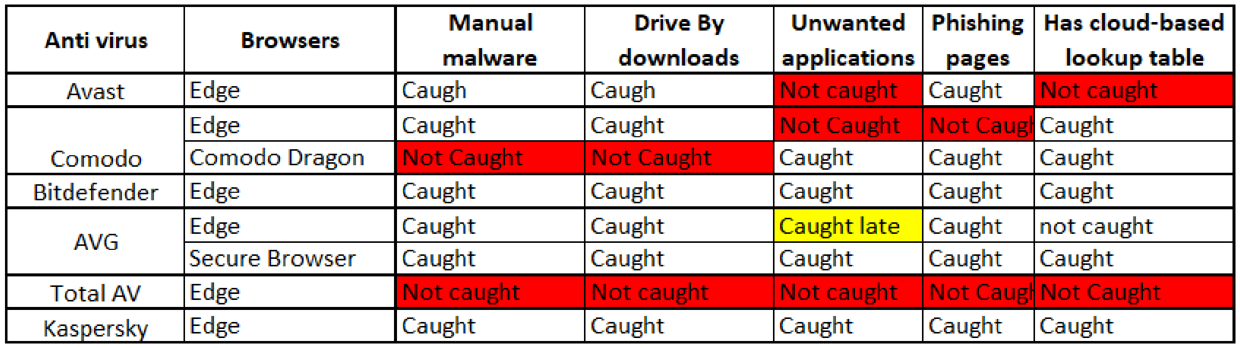
Table 1 showcases the results obtained after applying all the tests on the selected antivirus products. A ● indicates that the AV fully fulfills the characteristics definition, ◗ indicates a partial(quasi) implementation of the definition, a blank case indicates that the characteristic was not satisfied, and a "?" indicates that the feature could not be determined since it failed at D2 characteristics.

Table 1. Characteristic table results for antivirus products

AV Product	U1	U2	U3	U4	D1	D2	D3	S1	S2	S3	S4	S5
Avast	●	●	◗			●	◗	◗	◗	◗	●	●
Bitdefender	●	●	●			●	●	●	●	●	●	●
Total AV	◗		●			◗	●			●	●	◗
Comodo	●	●	◗		●	◗	●	◗	◗	◗	●	◗
AVG	●	●	◗		●	●	◗	◗	●	◗	●	●
Kaspersky	●	●	●			◗	●	●	●	●	●	●
PC Protector	?	?	?	?	?		?	?	?	?	?	?

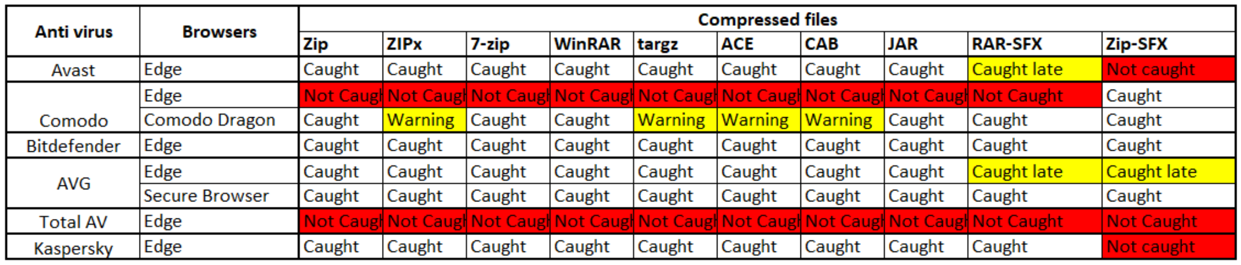
This table was generated after compiling the results from the different test procedures implemented. For the case of PC Protector, it required users to input a payment method to be able to run a free-trial version of their products, which we were not willing to give away for a temporary product. The AMTSO tests results can be found on table 2 and 3, which influenced the characteristic choices S1 and S2.

Table 2 AMTSO test on common user actions that lead to security faults.



Anti virus	Browsers	Manual malware	Drive By downloads	Unwanted applications	Phishing pages	Has cloud-based lookup table
Avast	Edge	Caugh	Caugh	Not caught	Caught	Not caught
Comodo	Edge	Caught	Caught	Not Caught	Not Caug	Caught
	Comodo Dragon	Not Caught	Not Caught	Caught	Caught	Caught
Bitdefender	Edge	Caught	Caught	Caught	Caught	Caught
AVG	Edge	Caught	Caught	Caught late	Caught	not caught
	Secure Browser	Caught	Caught	Caught	Caught	Caught
Total AV	Edge	Not caught	Not caught	Not caught	Not Caug	Not Caught
Kaspersky	Edge	Caught	Caught	Caught	Caught	Caught

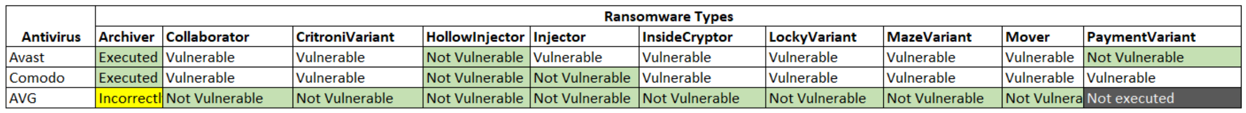
Table 3 AMTSO test on compressed files.



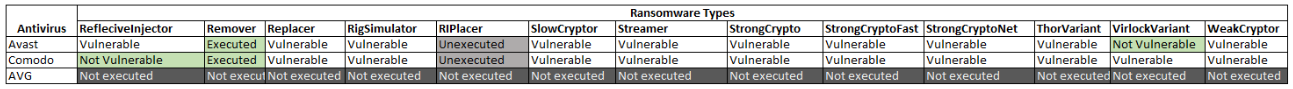
Anti virus	Browsers	Compressed files									
		Zip	ZIPx	7-zip	WinRAR	targz	ACE	CAB	JAR	RAR-SFX	Zip-SFX
Avast	Edge	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught late	Not caught
Comodo	Edge	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caught	Caught
	Comodo Dragon	Caught	Warning	Caught	Caught	Warning	Warning	Warning	Caught	Caught	Caught
Bitdefender	Edge	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught
AVG	Edge	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught late	Caught late
	Secure Browser	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught
Total AV	Edge	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caug	Not Caught	Not Caught
Kaspersky	Edge	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Caught	Not caught

As mentioned earlier in the paper, we compiled all the Ramsim reports into a single table, so that it would be easier to compare between the different products and their level of security against said attacks. Table 4 shows the results from running the Ramsim tool in the system. Total AV, Kaspersky, and Bitdefender are not included in the table, since they detected and destroyed the tests before they could be run. It is also worth noting that AVG did acted in a similar matter, but after giving the main tool the ok to execute, the tool ran its course contrary to the previous 3; this explains why AVG analysis has a significantly greater number of "Not Executed" tags as the AV destroyed the test files before an execution could be made.

Table 4. Ramsim test results



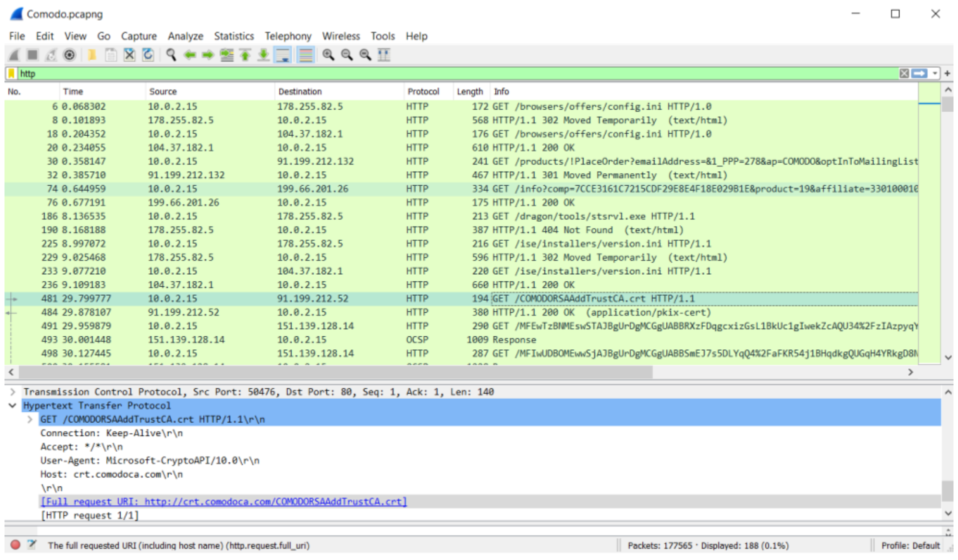
Antivirus	Ransomware Types									
	Archiver	Collaborator	CritroniVariant	HollowInjector	Injector	InsideCryptor	LockyVariant	MazeVariant	Mover	PaymentVariant
Avast	Executed	Vulnerable	Vulnerable	Not Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Not Vulnerable
Comodo	Executed	Vulnerable	Vulnerable	Not Vulnerable	Not Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable
AVG	Incorrectl	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnera	Not executed



Antivirus	Ransomware Types											
	RefleciveInjector	Remover	Replacer	RigSimulator	RIPlacer	SlowCryptor	Streamer	StrongCrypto	StrongCryptoFast	StrongCryptoNet	ThorVariant	VirlockVariant	WeakCryptor
Avast	Vulnerable	Executed	Vulnerable	Vulnerable	Unexecuted	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Not Vulnerable	Vulnerable
Comodo	Not Vulnerable	Executed	Vulnerable	Vulnerable	Unexecuted	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable
AVG	Not executed	Not execu	Not executed	Not executed	Not executed	Not executed	Not executed	Not executed	Not executed	Not executed	Not executed	Not executed	Not executed

Wireshark results discussion

In [1], HTTP-based connections are used between server connections during the updating of the Comodo antivirus program. We tested ourselves what kind of weak security vulnerabilities we could catch during this connection. Just as the book [1] uses Comodo, we used the Comodo program as well to understand how we can apply for other antivirus programs. During the update, we caught many packages through Wireshark that we needed to use the HTTP search to separate the packets we caught in our filter as shown in Figure 3. We tried to access the link that can be visible in HTTP connections under Hypertext Transfer Protocols.


Comodo.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time       Source          Destination     Protocol Length Info
    6 0.068302   10.0.2.15       178.255.82.5    HTTP      172 GET /browsers/offers/config.ini HTTP/1.0
    8 0.101893   178.255.82.5    10.0.2.15       HTTP      568 HTTP/1.1 302 Moved Temporarily  (text/html)
   18 0.204352   10.0.2.15       104.37.182.1    HTTP      176 GET /browsers/offers/config.ini HTTP/1.0
   20 0.234055   104.37.182.1    10.0.2.15       HTTP      610 HTTP/1.1 200 OK
   30 0.358147   10.0.2.15       91.199.212.132  HTTP      241 GET /products/!PlaceOrder?emailAddress=&1_PPP=278&ap=COMODO&optInToMailingList
   32 0.385710   91.199.212.132  10.0.2.15       HTTP      467 HTTP/1.1 301 Moved Permanently  (text/html)
   74 0.644959   10.0.2.15       199.66.201.26   HTTP      334 GET /info?comp=7CCE3161C7215CDF29E8E4F18E029B1E&product=19&affiliate=33010001
   76 0.677191   199.66.201.26   10.0.2.15       HTTP      175 HTTP/1.1 200 OK
  186 8.136535   10.0.2.15       178.255.82.5    HTTP      213 GET /dragon/tools/stsrvl.exe HTTP/1.1
  190 8.168188   178.255.82.5    10.0.2.15       HTTP      387 HTTP/1.1 404 Not Found  (text/html)
  225 8.997072   10.0.2.15       178.255.82.5    HTTP      216 GET /ise/installers/version.ini HTTP/1.1
  229 9.025468   178.255.82.5    10.0.2.15       HTTP      596 HTTP/1.1 302 Moved Temporarily  (text/html)
  233 9.077210   10.0.2.15       104.37.182.1    HTTP      220 GET /ise/installers/version.ini HTTP/1.1
  236 9.109183   104.37.182.1    10.0.2.15       HTTP      660 HTTP/1.1 200 OK
  481 29.799777  10.0.2.15       91.199.212.52   HTTP      194 GET /COMODORSAAddTrustCA.crt HTTP/1.1
  484 29.878107  91.199.212.52   10.0.2.15       HTTP      380 HTTP/1.1 200 OK  (application/pkix-cert)
  491 29.959879  10.0.2.15       151.139.128.14  HTTP      290 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBRXzFDqgcxizGsL1BkUc1gIwekZcAQU34%2FzIAzpyqY
  493 30.001448  151.139.128.14  10.0.2.15       OCSP     1009 Response
  498 30.127445  10.0.2.15       151.139.128.14  HTTP      287 GET /MFIwUDBOMEwwSjAJBgUrDgMCGgUABBSmEJ7s5DLYqQ4%2FaFKR54j1BHqdkgQUGqH4YRkgD8N

> Transmission Control Protocol, Src Port: 50476, Dst Port: 80, Seq: 1, Ack: 1, Len: 140
v Hypertext Transfer Protocol
  > GET /COMODORSAAddTrustCA.crt HTTP/1.1\r\n
    Connection: Keep-Alive\r\n
    Accept: */*\r\n
    User-Agent: Microsoft-CryptoAPI/10.0\r\n
    Host: crt.comodoca.com\r\n
    \r\n
    [Full request URI: http://crt.comodoca.com/COMODORSAAddTrustCA.crt]
    [HTTP request 1/1]

The full requested URI (including host name) (http.request.full_uri)    Packets: 177565 · Displayed: 188 (0.1%)    Profile: Default
```

Figure 3. Wireshark capture using Comodo

The book [1] suggests antiviruses use generally HTTP or HTTPS in rare case FTP for downloading signatures that the lists of downloadable files and remote relative URIs or full URLs. Also, the book [1] suggests some possibilities that an attacker can change a DNS record that the client will connect to the wrong IP address and download all the files without verifying. The other possibility is that an attacker can launch a man in the middle attack in a local area network and the attacker can modify the files during transit and supply bad copies of files or Trojanized versions of the antivirus products to the client machines. So, we focussed on the use of HTTP in update protocols in the antiviruses that open doors for possible attacks. Like the book [1] suggesting Comodo to analyze, we took Comodo as the first antivirus to start in Wireshark. You can see on the left side of Figure 4 that Wireshark shows a trace of a signature`s updating check. On the right side of Figure 4, we caught the HTTP through the URL of the update server above.

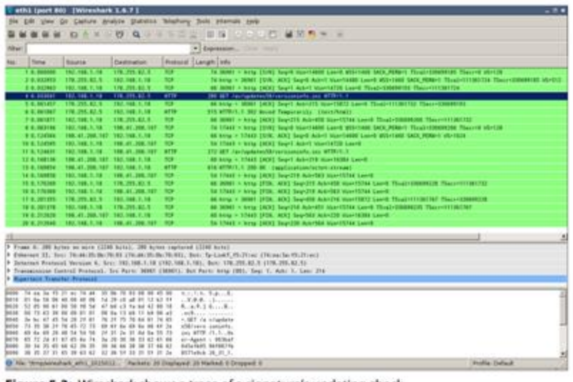


Figure 5-3: Wireshark shows a trace of a signature's updating check

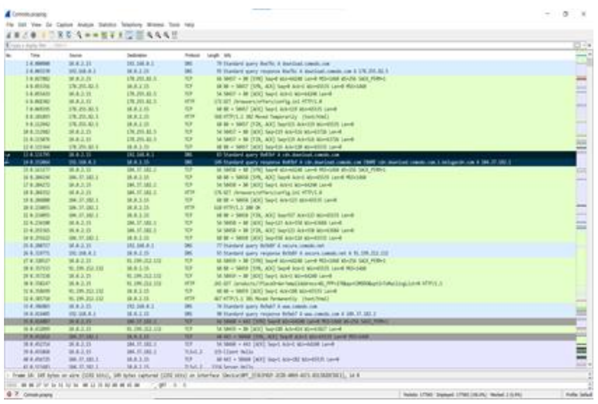
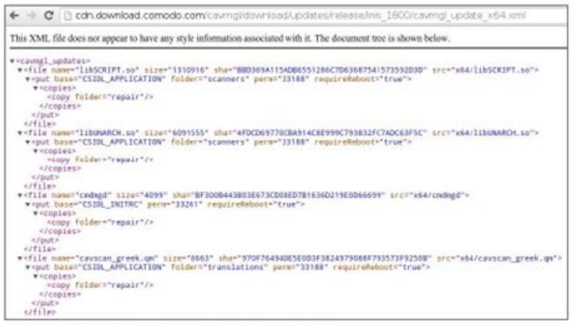


Figure 4 Wireshark comparison between [1] (left) and obtained (right)

According to the following URL on our side, our access has been rejected unlike the book [1] suggested. That is why we could not analyze the source code of the web site. That means Comodo improved their communication between client and web server for update downloading. Generally, we tried other various monitoring tools for analyzing the update system, but we did not get better results for our evaluation.



```
cdn.download.comodo.com/cavmgl/download/updates/release/inis_1800/cavmgl_update_x64.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

▼<cavmgl_updates>
  ▼<file name="libSCRIPT.so" size="1310916" sha="BBD369A115ADB6551286C7D6368754157359203D" src="x64/libSCRIPT.so">
    ▼<put base="CSIDL_APPLICATION" folder="scanners" perm="33188" requireReboot="true">
      ▼<copies>
          <copy folder="repair"/>
        </copies>
      </put>
    </file>
  ▼<file name="libUNARCH.so" size="6091555" sha="4FDCD69770CBA914C8E999C793832FC7ADC63F5C" src="x64/libUNARCH.so">
    ▼<put base="CSIDL_APPLICATION" folder="scanners" perm="33188" requireReboot="true">
      ▼<copies>
          <copy folder="repair"/>
        </copies>
      </put>
    </file>
  ▼<file name="cmdmgd" size="4099" sha="BF300B443B03E673CD08ED7B1636D219E0D66699" src="x64/cmdmgd">
    ▼<put base="CSIDL_INITRC" perm="33261" requireReboot="true">
      ▼<copies>
          <copy folder="repair"/>
        </copies>
      </put>
    </file>
  ▼<file name="cavscan_greek.qm" size="8663" sha="970F764940E5E003F3824979088F793573F9250B" src="x64/cavscan_greek.qm">
    ▼<put base="CSIDL_APPLICATION" folder="translations" perm="33188" requireReboot="true">
      ▼<copies>
          <copy folder="repair"/>
        </copies>
      </put>
    </file>
```

Figure 5-6: XML file to update Comodo software for Linux

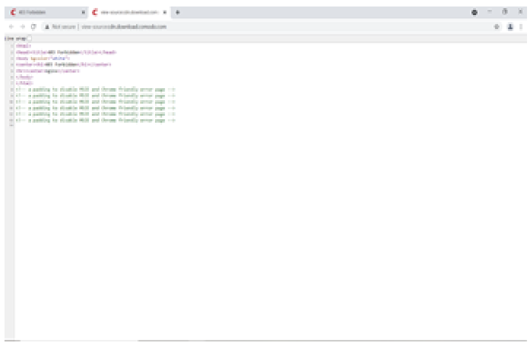


Figure 5. Access to site data comparison between [1] (left) and obtained (right)

Figure 5 shows a comparison between the Antivirus book`s [1] findings (on the left) and our findings (on the right) for the source code of the update server above. You can see on the left side Wireshark shows a trace of a signature`s updating check. On the right side we are caught through the URL of the update server.

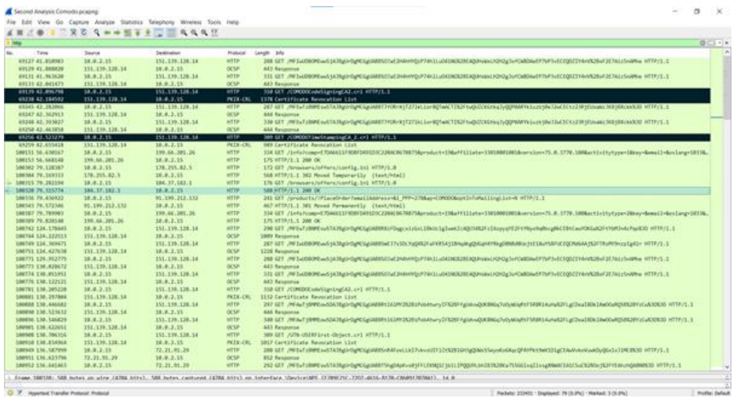
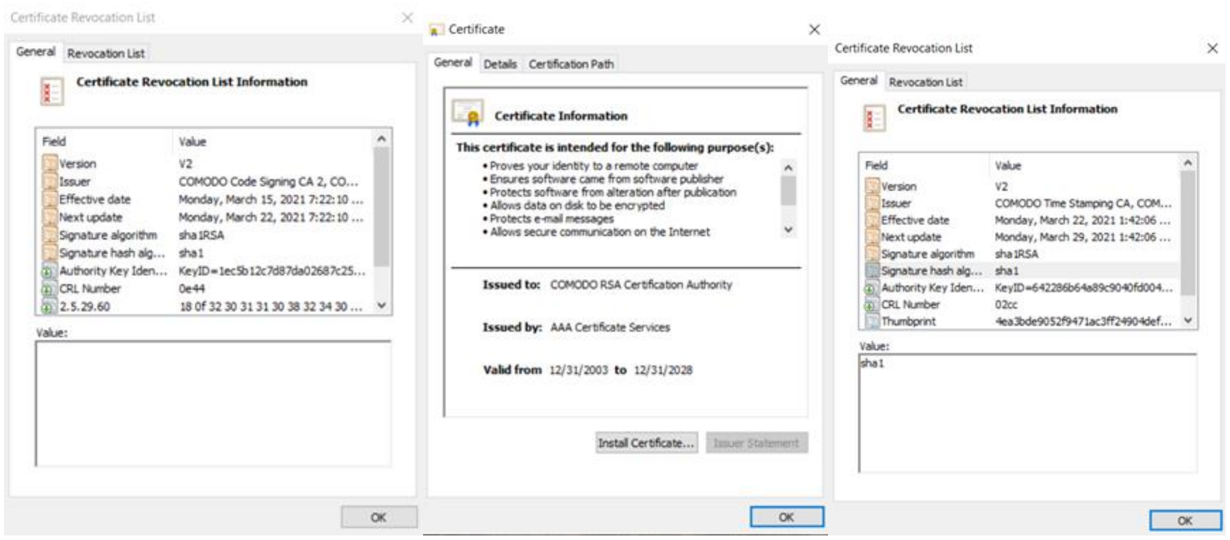


Figure 6. Obtaining packages through the links for Comodo's Certificates between client and server

Through the links that were founded in Wireshark, we downloaded COMODOCodeSigningCA, COMODORSAAddTrustCA, and COMODOTimeStampingCA. Also, we got Certificate Revocation related CAs. It can be seen in the picture below.



Certificate Revocation List

General | Revocation List

Certificate Revocation List Information

Field	Value
Version	V2
Issuer	COMODO Code Signing CA 2, CO...
Effective date	Monday, March 15, 2021 7:22:10 ...
Next update	Monday, March 22, 2021 7:22:10 ...
Signature algorithm	sha1RSA
Signature hash alg...	sha1
Authority Key Iden...	KeyID=1ec5b12c7d87da02687c25...
CRL Number	0e44
2.5.29.60	18 0f 32 30 31 31 30 38 32 34 30 ...

Value:

Certificate

General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):
- Proves your identity to a remote computer
- Ensures software came from software publisher
- Protects software from alteration after publication
- Allows data on disk to be encrypted
- Protects e-mail messages
- Allows secure communication on the Internet

Issued to: COMODO RSA Certification Authority

Issued by: AAA Certificate Services

Valid from 12/31/2003 to 12/31/2028

Install Certificate... | Issuer Statement

Certificate Revocation List

General | Revocation List

Certificate Revocation List Information

Field	Value
Version	V2
Issuer	COMODO Time Stamping CA, COM...
Effective date	Monday, March 22, 2021 1:42:06 ...
Next update	Monday, March 29, 2021 1:42:06 ...
Signature algorithm	sha1RSA
Signature hash alg...	sha1
Authority Key Iden...	KeyID=642286b64a89c9040fd004...
CRL Number	02cc
Thumbprint	4ea3bde9052f9471ac3ff24904def...

Value:
sha1

Figure 7. Comodo certificates obtained.

For other antiviruses, generally our access to the server for downloading was denied like Comodo or they used VPX style data file format that is used for Internet security application purposes. As you can see on the left side url caught during requests made to the AVG web servers to download updates. When we connected the web server, we got servers.def.vpx file. You can see in the picture on the right side that file content was secured. Interestingly, AVG antivirus was using an Avast antivirus-based data file when we checked what VPX meant.

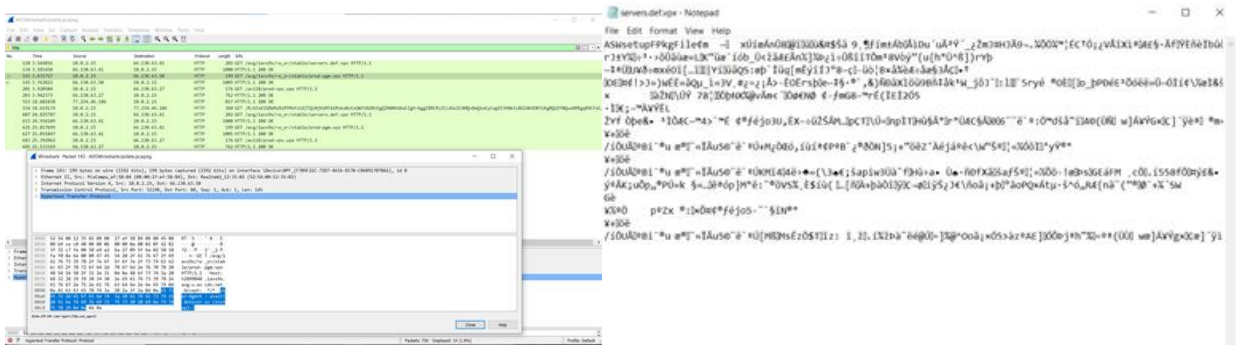
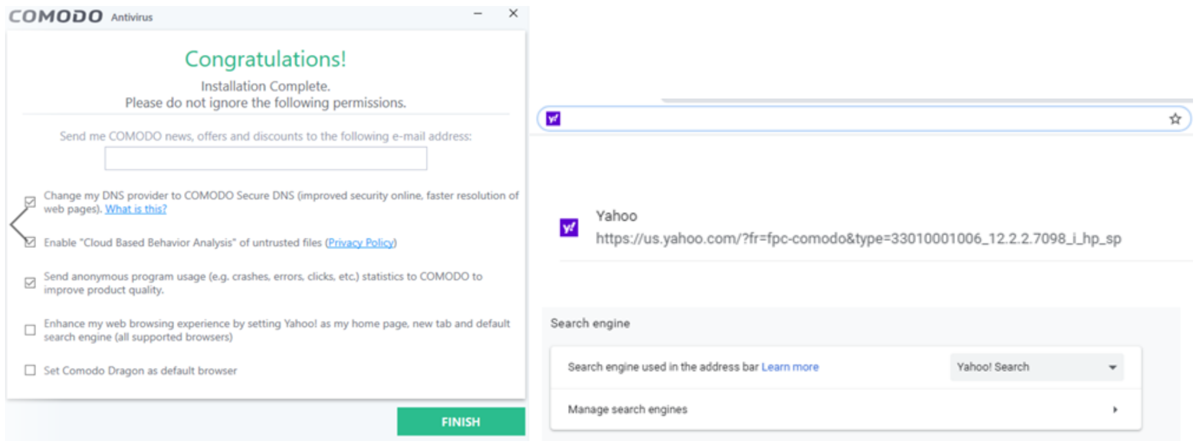


Figure 8. VPX file caught during Wireshark analysis of AVG in HTTP on the left side picture above and data file used by Avast, an antivirus and Internet security application(AVG Antivirus) on the right side picture above.

In addition, we have encountered some problems that will cause some security gaps that hit our eyes. These problems are the Yahoo browser problem and proxy script problem as seen in the pictures below.


COMODO Antivirus

Congratulations!
Installation Complete.
Please do not ignore the following permissions.

Send me COMODO news, offers and discounts to the following e-mail address:

- [✓] Change my DNS provider to COMODO Secure DNS (improved security online, faster resolution of web pages). What is this?
- [✓] Enable "Cloud Based Behavior Analysis" of untrusted files (Privacy Policy)
- [✓] Send anonymous program usage (e.g. crashes, errors, clicks, etc.) statistics to COMODO to improve product quality.
- [ ] Enhance my web browsing experience by setting Yahoo! as my home page, new tab and default search engine (all supported browsers)
- [ ] Set Comodo Dragon as default browser

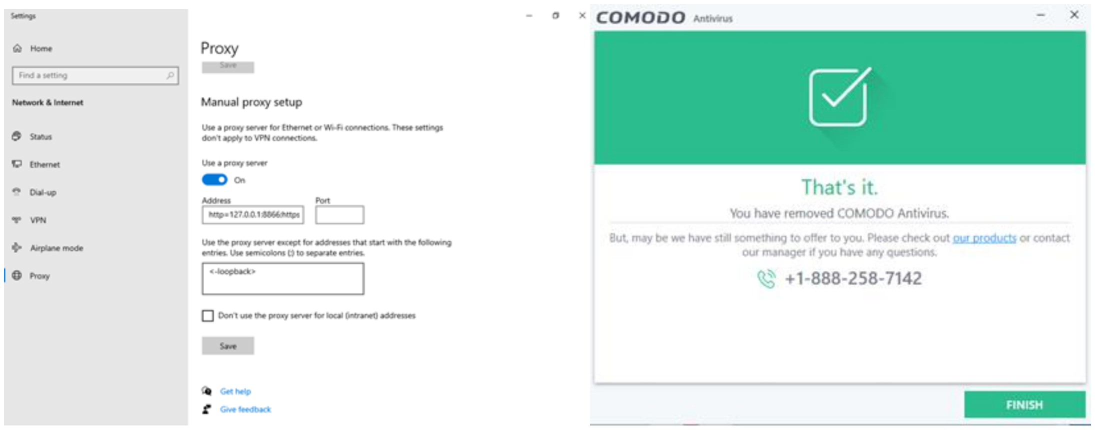
FINISH

Yahoo
https://us.yahoo.com/?fr=fpc-comodo&type=33010001006_12.2.2.7098_i_hp_sp

Search engine
Search engine used in the address bar Learn more: Yahoo! Search
Manage search engines

Figure 9. Yahoo browser configuration in Comodo

When it is necessary to mention Yahoo browser, as shown in the picture above, Yahoo browser can stay in the system even if it is not selected at the time of Comodo installation. Also, even when Comodo is deleted, Yahoo continues to stay as a default browser in the system. It can be seen in the picture on the left side that Yahoo browser stays in the home page, new tab, and as default search engine without choosing it at installation or after deletion of Comodo antivirus.



Proxy

Manual proxy setup

Use a proxy server for Ethernet or Wi-Fi connections. These settings don't apply to VPN connections.

Use a proxy server: On

Address: http=127.0.0.1:8866;https

Port:

Use the proxy server except for addresses that start with the following entries. Use semicolons (;) to separate entries.

<-loopback>

☐ Don't use the proxy server for local (intranet) addresses

That's it.

You have removed COMODO Antivirus.

But, may be we have still something to offer to you. Please check out our products or contact our manager if you have any questions.

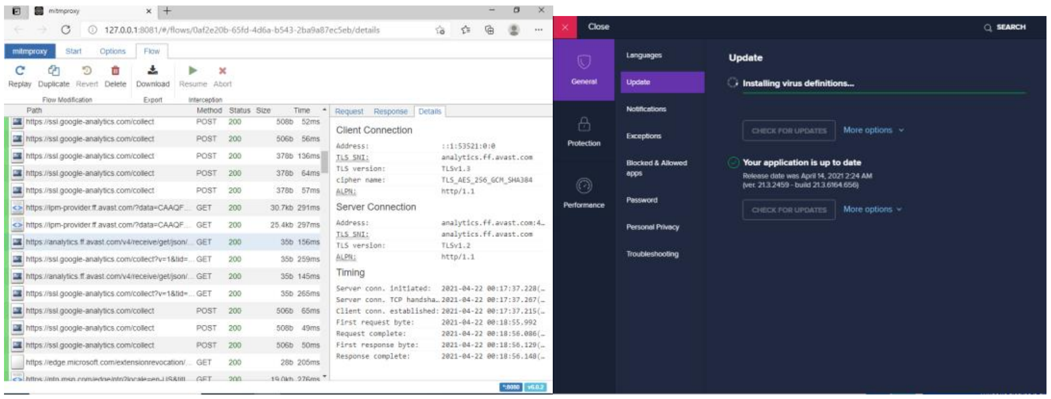
+1-888-258-7142

Figure 10. Proxy still active even after uninstalling Comodo

Speaking of proxy configuration, when Comodo antivirus is deleted from the system, the manually written script can continue to remain on the system. This may adversely affect the security of the system. Normally, all adjustments made by Comodo within the system should be deleted when expected. Users should not be expected to understand the advanced settings. Antivirus programs should securely arrange their automatic adjustments in this regard. While examining this situation, in our different trial we encountered, we saw that the script Comodo left back was affecting our internet connection and preventing our access to the internet. The script in the manual menu of proxy can be seen in the picture above that proxy configuration remains as changed after deletion of Comodo antivirus.

MITM Proxy discussion

As the pictures are shown below, Avast antivirus update server has been monitored through MitmProxy. While we are monitoring client-server communication, we tried updating the antivirus from the user side. Thanks to that, we got some https based packets that are sent to the user side. Also, we checked scanning features of images that made up look important and observed that they have been sent to the server side or third parties without using bit wise transmitted files. Generally, when we observed the size of an image that scanned through antivirus, it looked like 1KB or little more files. When they observed they were more bit wise, so we deduced there was no problem or violation for the privacy side in user images. But still some parts of information was sent to google analytic third parties so literally they were some shared information even the scanned images were bitwise and scrambled text.



```
127.0.0.1:8081/#/flows/0af2e20b-65fd-4d6a-b543-2ba9a87ec5eb/details

Path                                              Method Status Size   Time
https://ssl.google-analytics.com/collect          POST   200    508b   52ms
https://ssl.google-analytics.com/collect          POST   200    506b   56ms
https://ssl.google-analytics.com/collect          POST   200    378b  136ms
https://ssl.google-analytics.com/collect          POST   200    378b   64ms
https://ssl.google-analytics.com/collect          POST   200    378b   57ms
https://ipm-provider.ff.avast.com/?data=CAAQF...  GET    200  30.7kb  291ms
https://ipm-provider.ff.avast.com/?data=CAAQF...  GET    200  25.4kb  297ms
https://analytics.ff.avast.com/v4/receive/get/json/... GET 200  35b  156ms
https://ssl.google-analytics.com/collect?v=1&tid=... GET 200    35b  259ms
https://analytics.ff.avast.com/v4/receive/get/json/... GET 200  35b  145ms
https://ssl.google-analytics.com/collect?v=1&tid=... GET 200    35b  265ms
https://ssl.google-analytics.com/collect          POST   200    506b   65ms
https://ssl.google-analytics.com/collect          POST   200    508b   49ms
https://ssl.google-analytics.com/collect          POST   200    506b   50ms
https://edge.microsoft.com/extensionrevocation/... GET   200     28b  205ms
https://ntp.msn.com/edge/ntp?locale=en-US&titl... GET    200  19.0kb  276ms

Client Connection
Address:                  ::1:53521:0:0
TLS SNI:                  analytics.ff.avast.com
TLS version:              TLSv1.3
cipher name:              TLS_AES_256_GCM_SHA384
ALPN:                     http/1.1
Server Connection
Address:                  analytics.ff.avast.com:4…
TLS SNI:                  analytics.ff.avast.com
TLS version:              TLSv1.2
ALPN:                     http/1.1
Timing
Server conn. initiated:   2021-04-22 00:17:37.228(…
Server conn. TCP handsha… 2021-04-22 00:17:37.267(…
Client conn. established: 2021-04-22 00:17:37.215(…
First request byte:       2021-04-22 00:18:55.992
Request complete:         2021-04-22 00:18:56.086(…
First response byte:      2021-04-22 00:18:56.129(…
Response complete:        2021-04-22 00:18:56.140(…
```

Update

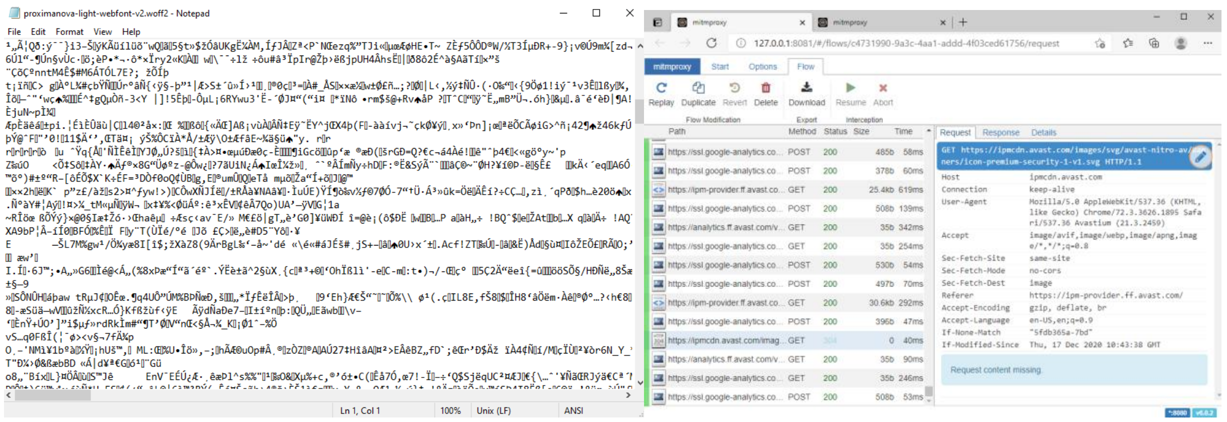
Installing virus definitions...

Your application is up to date

Release date was April 14, 2021 2:24 AM (ver. 21.3.2459 - build 21.3.6164.656)

Figure 11. Inspecting update of Avast using MITM proxy

The picture on the left below was taken during one of the https communications. We can see that the server protects itself even if trusted certificates are facilitated.



proximanova-light-webfont-v2.woff2 - Notepad

```
127.0.0.1:8081/#/flows/c4731990-9a3c-4aa1-addd-4f03ced61756/request

Path                                     Method Status Size   Time
https://ssl.google-analytics.co...       POST   200    485b   58ms
https://ssl.google-analytics.co...       POST   200    378b   60ms
https://ipm-provider.ff.avast.co...      GET    200  25.4kb  619ms
https://ssl.google-analytics.co...       POST   200    508b  139ms
https://analytics.ff.avast.com/v...      GET    200     35b  342ms
https://ssl.google-analytics.co...       GET    200     35b  254ms
https://ssl.google-analytics.co...       POST   200    530b   54ms
https://ssl.google-analytics.co...       POST   200    497b   70ms
https://ipm-provider.ff.avast.co...      GET    200  30.6kb  292ms
https://ssl.google-analytics.co...       POST   200    396b   47ms
https://ipmcdn.avast.com/imag...         GET    304       0   40ms
https://analytics.ff.avast.com/v...      GET    200     35b   90ms
https://ssl.google-analytics.co...       GET    200     35b  246ms
https://ssl.google-analytics.co...       POST   200    508b   53ms

GET https://ipmcdn.avast.com/images/svg/avast-nitro-av/ba
ners/icon-premium-security-1-v1.svg HTTP/1.1
Host                ipmcdn.avast.com
Connection          keep-alive
User-Agent          Mozilla/5.0 AppleWebKit/537.36 (KHTML,
                    like Gecko) Chrome/72.3.3626.1895 Safa
                    ri/537.36 Avastium (21.3.2459)
Accept              image/avif,image/webp,image/apng,imag
                    e/*,*/*;q=0.8
Sec-Fetch-Site      same-site
Sec-Fetch-Mode      no-cors
Sec-Fetch-Dest      image
Referer             https://ipm-provider.ff.avast.com/
Accept-Encoding     gzip, deflate, br
Accept-Language     en-US,en;q=0.9
If-None-Match       "5fdb365a-7bd"
If-Modified-Since   Thu, 17 Dec 2020 10:43:38 GMT

Request content missing.
```

Figure 12. File obtain during the inspection of the update of Avast

To sum up, the results of the analysis of all the antivirus were positive against security violations. Even though we used trusted certificates on our side, connections were denied or ceased during installation, update partially, and configuration. Some antiviruses even did not propose any packet transaction during monitoring like Comodo via MitmProxy.

Conclusion

In conclusion, applying the knowledge of the course, we were able to evaluate the security and privacy metrics of some popular antivirus. Through our analysis, we have found that some antiviruses share information to third party agents such as Google when running standard features, such as scanning and updating. We also found that the features marketed by said antivirus are in fact respected due to the wording, but there is clearly a difference in security level between them.

Our analysis could be improved in future works by adding obfuscation and real malware detection as recommended by the professor. We opted not to include said test due to time constraints and a necessity to create a new testing environment after discovering that our current testing environment has a forceful serial connection to the main system, which was recommended in several sources to be disable in order to have the highest level of security. We have attached in the following section different literature that we have encountered and read that helped in the creation of this report.

References

[1] J. Koret and E. Bachaalany, “The Antivirus Hacker's Handbook”, Indianapolis: John Wiley & Sons, Inc, 2015.

[2] S. Josse, "How to Assess the Effectiveness of Your Anti-virus?," Journal of Computer Virology 2, pp. 51-65, 2006.

[3] F. Hsu, M. Wu, C. Tso, C. Hsu and C. Chen, "Antivirus Software Shield Against Antivirus Terminators," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1439-1447, Oct. 2012, doi: 10.1109/TIFS.2012.2206028.

[4] N. Thamsirarak, T. Seethongchuen and P. Ratanaworabhan, "A case for malware that make antivirus irrelevant," *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Hua Hin, Thailand, 2015, pp. 1-6, doi: 10.1109/ECTICon.2015.7206972.

[5] J. Haffejee and B. Irwin, "Testing antivirus engines to determine their effectiveness as a security layer," *2014 Information Security for South Africa*, Johannesburg, South Africa, 2014, pp. 1-6, doi: 10.1109/ISSA.2014.6950496.

[6] J. Delaney, "The Effectiveness of Antivirus Software," ProQuest, New York, 2020.

[7] S. Zhu and et al., "Measuring and Modeling the Label Dynamics of Online Anti-Malware Engines," in 2020 USENIX Security Syposium, Boston, MA, 2020.

Appendix 1: Ransomware simulations

Archiver	Simply archives files using the gzip algorithm. This scenario should not be blocked.
Collaborator	Encrypts files similarly to a common version of Critroni. However, it relies on different processes for file enumeration, movement and deletion.
CritroniVariant	Simulates the behaviour of a common version of Critroni ransomware.
HollowInjector	Encrypts files by injecting the encryption code into a legitimate process using process hollowing.
Injector	Encrypts files by injecting the encryption code into a legitimate process using a common approach.
InsideCryptor	Encrypts files using strong encryption and overwrites most of the content of the original files with the encrypted data.
LockyVariant	Simulates the file activity performed by a popular version of Locky ransomware.
MazeVariant	Simulates the file related operations performed by Maze ransomware.
Mover	Encrypts files in a different folder using strong encryption and safely deletes the original files.
PaymerVariant	Simulates file related operations performed by DoppelPaymer-like ransomware.
ReflectiveInjector	Encrypts files by injecting the encryption code into a legitimate process using an advanced approach.
Remover	Simply deletes files and does not create any file. This scenario should not be blocked.
Replacer	Replaces the content of the original files. A real ransomware would show a message that fools users into thinking they can recover them.
RigSimulator	Simulates a mining rig which uses the machine CPU to mine Monero.
RIPlacer	Attempts to encrypt files in a folder subject to controlled folder access ransomware protection. This scenario requires special stings. Check the help page for details.
SlowCryptor	Simulates the behaviour of a ransomware variant that encrypts files slowly, to avoid detection by security products.

Streamer	Encrypts files and writes data into a single file, using strong encryption, then deletes the original files.
StrongCryptor	Encrypts files using strong encryption and safely deletes the original files.
StrongCryptorFast	Encrypts files using strong encryption and deletes the original files. It also simulates sending the encryption key to a server using a HTTP connection.
ThorVariant	Simulates file related operations performed by Thor ransomware.
WeakCryptor	Encrypts files using weak encryption and deletes the original files.
Virlock	Simulates file related activity of a common version of Virlock ransomware.

Appendix 2: AMTSO Test descriptions

Detect manually downloaded malware.

This test downloads attempts to download the EICAR test file, which was developed by European Institute for Computer Antivirus Research (EICAR) and Computer Antivirus Research Organization(CARO) to test the response of computer antivirus programs. This file is not inherently malicious, but it will demonstrate that the installed AV is configured correctly and conforms to industry best practices. Should this file be downloaded, it demonstrates that the AV is not satisfying one of the previous 2 conditions.

Detect potentially Unwanted Applications (PUAs).

Much like the previous test, this test will attempt to download a “steril” Potentially Unwanted Application, or Potentially Unwanted Software, to the local machine. The anti malware should be able to identify this file as such and block the download of the file. Failure to block the download will show that there are some configurations missing in the antivirus to detect these sorts of files from being downloaded.

Detect drive-by downloads of malware.

Drive-by downloads refer to the unintended download of computer software from the internet which can occur in 2 ways:

1. The person has authorized the download without understanding the consequences automatically.

2. The download occurs without the user's knowledge which are often used by computer viruses, spywares, malware, or crimeware.

This test will simulate such downloads by activating a separate tab on the browser and attempting to download the EICAR file previously mentioned. A successful download would indicate that the AV is not configured correctly or that it does not abide by industry best practices.

Detect phishing pages.

This test will attempt to open a webpage that is considered by industry to be a phishing page. Should the web page be loaded, it would indicate that the AV does not yet support this FSC or that the AV's Anti-Phishing feature is not configured properly or not enabled.

Detect compressed malware.

This test will try to download different compressed file formats containing the EICAR file. Should a file type download successfully, it would indicate that the AV is not configured properly, does not support handling selected compression methods, or does not abide by industry best practices.

Is connected to a cloud-based lookup system.

Much like previous SFC tools, this test will attempt to download a test file which is only detected by querying a cloud-based reputation system. As such, being able to download this file would indicate that the AV does not have a cloud-base look up or does not abide by industry best practice.