Masquerade Attacks Against Security Software Exclusion Lists

Timothy McIntosh[1], Julian Jang-Jaccard[1], Paul Watters[2], and Teo Susnjak[1]

[1] Massey University, Auckland 0632 New Zealand
t.mcintosh@massey.ac.nz
[2] La Trobe University, Bundoora VIC 3086, Australia

Abstract. Security software, commonly known as Antivirus, has evolved from simple virus scanners to become multi-functional security suites. To combat ever-growing malware threats, modern security software utilizes both static and dynamic analysis to assess malware threats, inevitably leading to occasional false positive and false negative reports. To mitigate this, existing state-of-the-art security software offers the feature of *Exclusion Lists* to allow users to exclude specified files and folders from being scanned or monitored. Through rigorous evaluation, however, we found that some of such products stored their *Exclusion Lists* as unencrypted cleartexts either in known or predictable locations. In this paper we empirically demonstrate how easy it is to exploit the *Exclusion Lists* by launching masquerade attacks. We argue that the *Exclusion Lists* should be better implemented such as using application whitelisting, the contents of the lists to be better safeguarded, and only be readable by authorized entities within a strong access control scheme.

Keywords: Antivirus · Security Software · Exclusion List · Masquerade Attack.

1 Introduction

Malware has been a major threat to information security since its early inception. As one of the most damaging types of malware, ransomware has recently drawn much attention due to the unprecedented infiltration with disastrous consequences[12]. To combat the threat of malware, especially ransomware, computer users are often advised to install powerful and sophisticated security software with anti-malware functionality, to perform regular backups, and to take precautions when executing unknown programs. Security software is typically a suite of different packages that can protect computer systems from malware, intrusions and other security exploits, to deliver a high level of assurance [2]. As security software becomes more powerful and sophisticated, malware has to either evade detection, or disrupt the functionality of security software in any successful malware campaign [8]. In 2007, approximately 10% of malware was found to be able to disrupt the functionality of security software[23]. However, since Windows 8.1 in 2013, Microsoft introduced the concept of *Protected Anti-Malware Services*, to protect user-mode services of security software, making attacks by malware more difficult [10]. To evade detection, malware developers have employed various approaches, including polymorphism, obfuscation and fileless scripting attacks [2, 8].

Security software employs a combination of static and dynamic analysis to analyze suspicious files, resulting in occasional false positive or false negative reports [2]. To address the issue of false positives, almost all security software products provide a set of *Exclusion Lists* to allow experienced users to manually mark suspicious programs detected by security software as *permitted* or *ignored*. Sometimes the *Exclusion List* is also used to exclude folders like *System Volume Information* to speed up the threat scanning process. The *Exclusion List* is a simpler implementation of application whitelisting often with only specifications on file names and absolute file paths. It is typically achieved by either asking the user to either permit the application during its runtime, or allowing users to later add selected applications to the *Exclusion List* in software settings. The *Exclusion Lists* rely on users being competent to implement such a list and, may be more appropriate in tightly controlled corporate environments where security takes precedence over convenience and competent system administrators are available [5, 6].

Problem statement. We believe some popular security software products are vulnerable to masquerade attacks by malware exploiting their *Exclusion Lists*. A masquerade attack occurs when an attacker (masquerader) steals the identity of a legitimate user to gain access to whatever the victim has authorization [11]. We believe the *Exclusion Lists* of some security software have not been implemented property to prevent unauthorized access. Our hypothesis is that it is possible for unknown malware to exploit the *Exclusion Lists* of security software and masquerade as excluded applications to launch destructive attacks, while being completely ignored by security software.

Summary of Original Contributions

- We found that 8 out of the 10 popular security software products we examined stored their *Exclusion Lists* as unencrypted cleartext and their records could be programmatically obtained.

– We performed proof-of-concept masquerade attacks on those 10 security software products by exploiting their *Exclusion Lists*, and explained how the existing defense mechanisms of some security software products could be bypassed.
– We disclosed this issue responsibly to affected security software vendors and obtained their responses.
– We proposed how the *Exclusion Lists* mechanism of security software could be better implemented to close this apparent security loophole.

2 Related Work

Malware could actively engage with security software to probe its functionality or interrupt its security protection. Studies have been conducted to demonstrate how to programmatically and systematically explore or exploit security software engines. In [20], a testing framework was developed to automatically test and explore capabilities of 50 antivirus engines in a black-box approach. It was concluded that applying mutations to input malware samples could trigger heuristic analysis and affect the detection outcome. In [6], a known malware binary was deployed in combination with other evasive techniques to test the detection of numerous security software products. Some malware samples were found to interfere or terminate the functionality of security software [8]. In [14], some security software products were found to temporarily suspend real-time monitoring service during routine updates; some malware could trigger security software updates and attack during the vulnerable update window when real-time monitoring is temporarily suspended. In [13], the researchers attempted to exploit the self-protection of 12 security software products, and found that many of them were subject to subversion by malware.

Some modern malware passively implements sophisticated code obfuscation techniques to evade detection [18]. Malware like *Trojan.Spyeye* is staged malware that uses a dropper to deliver the payloads in stages, to reduce the exposure of malicious code, hide true functionality, and minimize the risk of being detected by security software [14]. Some malware utilizes extensive encryption and compression packers to obfuscate their malicious code, making it difficult for security software to conduct syntactic or semantic analysis [15]. Some malware searches for unique system traits or artifacts to speculate on the presence of security software, subsequently behaving in a benign manner or even self-terminate to evade detection [4]. In [19], it was found that applying Return-Oriented Programming (ROP) could achieve comprehensive polymorphism and evade detection of most AntiVirus software, by transforming malicious code to its ROP equivalent. In [24], several evasion techniques were combined to enable some known malware to become undetectable by most security software products. Dynamic analysis with machine learning could unveil real malware activities, and how they interact with operating systems, but some malware has developed mechanisms to obscure their behaviors either by making irrelevant API calls in between attacks, or by detecting the presence of virtualized environments and remaining dormant [15].

3 Usage of Exclusion Lists

Many security vendors warn users that defining exclusions could lower the protection offered by security software, and recommend it should be exercised with caution only by experienced users who are confident that excluded files are not malicious. In this section, the usage of *Exclusion Lists* in private and corporate settings is explored.

3.1 In Private Settings

In private settings, individuals tend to use *Exclusion Lists* to bypass software restrictions imposed by security software to execute certain applications, either intentionally or by deception.

To Execute Keygen and Other Applications Reported by Security Software Many security software products report keygen and similar applications as malicious or potentially unsafe. For example, the "AMT Emulator" is a universal Adobe™ product patcher currently circulating on the Internet. The version 0.9.2 is reported to be malicious by 39/71 scanning engines on VirusTotal in January 2019. Avast, AVG and Kaspersky reported it as clean whereas Bitdefender, ESET, McAfee, Microsoft, Sophos and Symantec considered it potentially unsafe. The "Office 2013 - 2019 C2R Install" is a hacking tool used to crack Microsoft Office products. It was reported on VirusTotal to be unsafe by 39/71 scanning engines, including AVG, BitDefender, ESET, McAfee, Microsoft, Sophos, Symantec and Trend Micro. Users who wish to crack their illegal copies of commercial software would have to add those applications to the *Exclusion List* of their security software to enable execution.

Due to Deception by Phishing and Other Malware Many phishing sites either distribute legitimate software with malicious binaries, or fake software in the disguise of media players for pirated movies [25] or fake readers for pirated eBooks [26]. They pretend to warm users that their software distribution is legitimate, and entice users to ignore warnings from security software about the possible threat. For example, Format Factory is a free and multi-functional multimedia file converter, but it contains the Adware *DealPly* in its installation package, which is considered malicious by ESET Smart Security. Unsophisticated users, being enticed with free software or pirated media content, can be tricked with social engineering techniques into temporarily disabling their security software or excluding malware from scanning, and can become victims of malware attacks [26].

3.2 In Corporate Settings

In corporate settings, improving system stability or optimizing performance is of high significance [3]. Files excluded are usually non-executable, temporary or usually of lower risk of malware infection [3, 7, 9].

To Improve System Stability This is a commonly cited purpose for which to implement such an *Exclusion List*, especially in production environments [1, 9]. Many commercial software vendors have recommended Antivirus exclusions to prevent their products from mistakenly being identified as malware (*i.e.* false positive) by security software, which could interfere or interrupt with their normal functionalities. For example, CurrentWare™, a web-filtering and employee monitoring software, recommends that its monitoring agent, client end and the helper program be added to Antivirus *Exclusion Lists* [1]. Such commercial software products are often required to be running constantly on servers to guarantee almost 24/7 availability. If they are marked as false positive and terminated by security software after signature updates, it could lead to disastrous consequences in production environments.

To Optimize System Performance Another possible reason to implement *Exclusion Lists* in corporate settings is the need to optimize system performance by excluding items that system administrators do not find necessary to be scanned regularly by security software. This is because many clients expressed concerns of possible performance degradation of time-critical systems due to the deployment of security software [5]. Microsoft recommends that on host operating systems running Hyper-V virtual machines, it is best practice to exclude virtual hard-disk files (VHD, VHDX *etc.*), virtual machine configuration directory, snapshot file directory, Vmms.exe (Virtual Machine Management Service) and Vmwp.exe (Virtual Machine Worker Process). As many experienced system administrators only run applications inside the virtualized guest OS, the attack surface of host OS being infected by malware is further reduced. [9]. [7] recommended using exclusions on virtual machine files, subversion databases, photos, music and Windows Update folders to resolve performance issues.

3.3 Risk of Implementing Exclusion Lists

Some defensive security researchers are concerned about the risk of implementing *Exclusion Lists* for security software. [22] commented on the official exclusion list recommendations from Microsoft (the updated 2019 version in [3]), and was concerned about the security risks brought by the recommendations. According to [17], because the recommended Antivirus exclusion lists were widely available to the public, malware developers could use whitelisted files to deliver Advanced Persistent Threat (APTs). This could target infection groups, by inserting their malware into the file exclusion folders or to occasionally force the Antivirus configuration to exclude the specified malware files. [17] recommended that organizations should implement multi-layered defense in depth and should not rely solely on security software products; users should fully understand the risks and should avoid doing so unless there is a critical reason.

4 Proof-of-Concept Masquerade Attack by Replacing Excluded Programs

In this section, a proof-of-concept masquerade attack is demonstrated by obtaining the *Exclusion Lists* of the security software programatically by parsing the list and replacing program executables on the list.

4.1 Sample Selection

We carefully chose a selection of full-functionality retail security software for personal computers (Table 1) based on their perceived popularity, market share and the availability of trial versions [16, 21]. Each security software was

installed individually to the default path with all possible modules enabled at default settings, and had its antivirus definitions and program modules updated after initial installation, on the day of January 8, 2019. A few security software products provided additional features to combat against ransomware encryption on user files, and those modules were enabled whenever provided. We obtained a sample of *Ransom.WannaCry* as known ransomware. We implemented a simple program as the unknown ransomware to iterate "Documents" folder, encrypt all DOCX and PDF files using the Windows built-in *RijndaelManaged* cryptographic class, and sleep for 5 seconds between each file encryption.

Table 1. List of Security Software Tested

Software Vendor	Product Name	Product Version	Anti-Ransomware Module
Avast	Avast Internet Security	19.1.2360	
AVG	AVG Internet Security	19.1.3075	Yes
BitDefender	Bitdefender Total Security	23.0.16.72	Yes
ESET	ESET Smart Security	12.0.31.0	(requires online ESET LiveGrid)
Kaspersky	Kaspersky Internet Security	19.0.0.1088	
McAfee	McAfee Total Protection	16.0 R14	
Microsoft	Windows Defender	1.283.2487.0	Yes
Sophos	Sophos Home Premium	1.3.3	
Symantec	Norton Security Premium	22.16.3.21	
Trend Micro	Trend Micro Maximum Security	15.0.1212	Yes

4.2 Steps to Perform Masquerade Attacks

We followed the steps below to simulate masquerade attacks by both known and unknown ransomware samples.

1. Scan the ransomware sample with the security software.
   - If the unknown sample cannot be identified, execute it to check whether its attack can be thwarted.
2. Add "WINWORD.EXE" (Microsoft Word) to all exclusion lists of security software
3. Use Word to modify any docx file in "Documents" folder.
4. Go to the installation folder of "WINWORD.EXE" and replace it with a ransomware renamed as "WINWORD.exe".
   - The real-time protection of the security software has to be temporarily disabled to allow the copying of *WannaCry* sample to overwrite "WINWORD.exe". Disabling real-time protection is not required for the unknown ransomware sample.
5. Execute the fake "WINWORD.EXE" (ransomware).

4.3 Storage of the Exclusion Lists

We investigated the possible locations of storage of the explicit *Exclusion Lists* of security software in Windows Registry, in the installation folder of the security software in "Program Files", in "ProgramData" and in the "AppData" folder of the user profile. We summarized the results in Table 2. We found that Avast Internet Security, AVG Internet Security, Bitdefender Total Security and Kaspersky Internet Security stored the lists in unencrypted cleartext configuration files. McAfee Total Protection, Microsoft Windows Defender and Trend Micro Maximum Security stored the lists in Windows Registry values. Sophos Home Premium used a web portal for software configuration and cached the settings locally; users could not change software settings without live Internet connections. We did not find Windows Registry values or configuration files of ESET Internet Security or Symantec Norton Security Premium.

Table 2. Storage of the Exclusion Lists

Product Name	General Exclusion Lists	Anti-Ransomware Exclusion Lists
Avast Internet Security	C:\ProgramData\AVAST Software \Avast\exclusions.ini	N/A
AVG Internet Security	C:\ProgramData\AVG\Antivirus \exclusions.ini	C:\ProgramData\AVG\Antivirus \exclusions.ini
Bitdefender Total Security	C:\Program Files\Bitdefender \Bitdefender Security\settings\system \excludemgr.xml ; C:\Program Files\Bitdefender \Bitdefender Security\settings\system \LGKC\ExcludeMgr.xml	C:\Program Files\Bitdefender \Bitdefender Security\settings\bdrsp.xml
ESET Smart Security	Cleartext list not found	Possibly cloud-based by ESET LiveGrid
Kaspersky Internet Security	C:\ProgramData\Kaspersky Lab \AVP19.0.0\Data\settings_kis.kvdb-wal	N/A
McAfee Total Protection	HKEY_LOCAL_MACHINE\SOFTWARE \McAfee\VirusScan\Overrides\System \Settings\VSO\OAS	N/A
Microsoft Windows Defender	HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows Defender\Exclusions \Paths	HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows Defender \Windows Defender Exploit Guard \Controlled Folder Access \AllowedApplications
Sophos Home Premium	C:\ProgramData\Sophos Management Communications System \Endpoint\Cache\SAV.status (cached from web portal)	N/A
Symantec Norton Security Premium	Cleartext list not found	N/A
Trend Micro Maximum Security	HKEY_LOCAL_MACHINE\SOFTWARE \TrendMicro\UniClient\1700\Scan \Exceptions	HKEY_LOCAL_MACHINE\SOFTWARE \TrendMicro\UniClient\1700\FolderShield \UserExceptions

4.4 Masquerade Attacks by the Unknown Ransomware

Our proof-of-concept ransomware sample was able to encrypt files in "Documents" folder on all systems monitored by security software that did not have offline anti-ransomware modules. Of the four security software products that offered anti-ransomware modules, we were able to evade detection by Trend Micro Maximum Security by only partially encrypting PDF files and sleeping 5 seconds between each file encryption. Avast Internet Security, AVG Internet Security and Bitdefender Total Security appeared to operate anti-ransomware defense in a strict way that required prior application whitelisting, and blocked file access by our proof-of-concept ransomware sample.

We later added "WINWORD.EXE" to all general *Exclusion Lists* of security software and the anti-ransomware *Exclusion Lists* of those four security software, replaced "WINWORD.EXE" executable with our proof-of-concept ransomware sample renamed as "WINWORD.EXE", and executed the fake "WINWORD.EXE" in the new file path. None of the real-time protection of security software noticed the file change, and our proof-of-concept ransomware sample was able to encrypt files in "Documents" folder as designed.

4.5 Masquerade Attacks by Ransom.WannaCry

All security software products tested were able to correctly identify our known ransomware as WannaCry or WannaCryptor, and were able to remove the ransomware executable before it could be executed. However, when "WINWORD.EXE" was added to the *Exclusion List* and the actual application executable was later replaced by WannaCry sample, none of the security software reported the masquerading behavior nor reported the application as WannaCry upon its execution. Due to the self-extracting nature of this WannaCry sample, the assistant files taskdl.exe and taskse.exe extracted to the same directory were correctly identified as WannaCryptor modules and were removed by security software. Nevertheless, the main component of WannaCry payload masqueraded as "WINWORD.EXE" still managed to encrypt user files in "Documents" folder.

5 Responsible Disclosure

Using the practice of responsible disclosure, we summarized our findings and disclosed to the seven companies which stored the *Exclusion List* in cleartext formats. Sophos was not included in the list of security software vendors, as its Bug Bounty program explicitly prohibited disclosure to the public after an issue was submitted to them. The responses from security software vendors to our responsible disclosure at the time of this paper are listed in Table 3.

Table 3. Response from Security Software Vendors to Our Responsible Disclosure

Software Vendor	**Assessed Severity**	**Status**	**Comments**
Avast	-	In Progress	Avast thought there was a risk of deploying an APT in the system in this manner.
AVG	-	Acknowledged	AVG received the email and acknowledged the possibility of such an attack.
Bitdefender	-	Acknowledged	Information has been forwarded to their response team. No additional comments were made.
Kaspersky	Low	Resolved	The issue will be addressed in a future version.
McAfee	Medium Low	In Progress	Their architects are working through possible solutions to remove the plain-text entries from the registry.
Microsoft	-	Closed as Non-Issue	Their engineers investigated the issue and determined the behavior was by design.
Trend Micro	-	Closed as Non-Issue	They didn't think the exclusion list issue was relevant.

Among the vendors that have responded, Microsoft and Trend Micro consider it a non-issue.

> *"Thank you for your submission. Our engineers investigated your report and determined the behavior you described is by design."* (Microsoft Security Response Center)

> *"The exclusion list issue is irrelevant for a file that we do not detect ... Creating you own files and calling them 'malware' is not a typical practice to show vulnerabilities."* (Trend Micro)

Avast, Kaspersky and McAfee acknowledged there could be a low risk when the *Exclusion Lists* were exploited.

> *"We are aware of this possibility... in general, exclusions are dangerous and we don't recommend users to use them, other than temporarily when they encounter a false positive. Historically, we also tried to make adding an exclusion 'not too user friendly' so that it cannot be done too easily or even by mistake, but it's always a compromise between users asking for the feature (for whatever reason) and the security point of view. ... But yes, it can be useful as a persistence method"* (Avast)

> *"This issue may be classified as AV Bypass, but its real severity is Low or Moderate. ... If an item from exclusions points to a location with a strong ACL, an attacker should have admin privileges to overwrite it ... We plan to release a fix for this vulnerability in the following releases of the product"* (Kaspersky)

While some security software vendors disregarded our findings, others still acknowledged the possibility of such attacks and were actively planning product changes. We believe a loophole has been found that could be used to compromise the security of protected systems. The feature of *Exclusion List* is provided to users as a workaround to address the issue of false positive detection by security software, which is almost impossible to avoid completely[4]. Security software solutions sold to the market to protect them from malware attacks are also available to cybercriminals. As soon as users implement such a list and the list is retrievable by cybercriminals, it is possible for cybercriminals to experiment with the security software and exploit possible security loopholes, such as the attack via *Exclusion List* described above.

6 Discussion

No academic research has been found to investigate the pros and cons of implementing security software *Exclusion Lists*. However, such lists are commonly implemented in practice, due to false positive rates or impact on system performance by security software. While some industry researchers raised general concerns on the possibility of exploiting the *Exclusion Lists*, we demonstrated that some security software implemented the lists as clear-text in known or predictable locations, possibly making such exploits easier to be conducted. Despite that, no consensus existed in the security software industry regarding the nature and severity of the issue reported. There has been no known or reported exploits on security software *Exclusion Lists*. However, we believe it is a design vulnerability and should be addressed.

To rectify this design vulnerability, we recommend that the *Exclusion Lists* of security software should be safeguarded with confidentiality, and should implement proper application whitelisting that combines both the absolute paths of the applications and the cryptographic hashes of the application executables to protect its integrity. We found that many commercial applications were properly signed with valid certificates and their publishers usually did not change when the products were updated or patched. As a result, we recommend the following decision-making process to control the admission of whitelisted applications. When an application is added to the *Exclusion Lists*, its cryptographic hash and publisher certificate (if available) should be obtained and stored by the security software. When the application executable has changed, if it is signed by the same publisher, the file change could be considered a software update, and the security software would re-compute and update the cryptographic hash value. In all other scenarios, the file change should require approval from experienced users or system administrators. We believe a similar algorithm is possibly already implemented by some security software. For example, ESET Smart Security asks for user permission to grant access to webcam each time after each Google Chrome or Skype update. We also recommend that the *Exclusion Lists* of security software should be stored in non-cleartext formats in locations unknown to the public, as Norton Security Premium and ESET Smart Security did, so malware cannot easily read them to identify applications on the *Excluded Lists*.

7 Conclusion

In this paper, we examined how the *Exclusion Lists* of popular security software were implemented. We found that 8 out of 10 popular security software we examined stored their *Exclusion Lists* in unencrypted cleartext formats, either as values in Windows Registry, or as configuration files on the hard drive. We found that the cleartext values of *Exclusion Lists* could be read programmatically. We also found that when an application executable on the *Exclusion Lists* was replaced by a malware, the security software did not detect abnormalities unless the malware was already known by the security software. Furthermore, when a malware was executed masquerading as an application on the *Exclusion Lists*, even if the malware was known to the security software, the malicious behaviors were ignored. By exploring proof-of-concept attack vectors on the *Exclusion List* of a few security software products, we were able to reveal the design vulnerability that could give malware an "All Access" exemption to evade detection and attack operating systems. However, after we performed responsible disclosure to security vendors, only some of them have acknowledged this issue and promised to address it in future releases.

We believe more work should be done to properly implement *Exclusion Lists* applying the principles of secure application whitelisting to maintain confidentiality of the list and the integrity of the whitelisting mechanism. Security software should monitor changes of application executables included in the *Exclusion Lists* and, store the lists either as encrypted configuration files on the disk or as one-way cryptographic hash values in Windows Registry.

Acknowledgment

This work was made possible by the support of a grant (UOCX1720) from the Ministry of Business, Innovation and Employment of New Zealand, September 2017 Catalyst: Strategic Investment Round.

References

1. Adding currentware files to your antivirus' exclusion list, `https://www.currentware.com/faqs/av-exclusion-list/`
2. Al-rimy, B.A.S., Maarof, M.A., Shaid, S.Z.M.: Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. Computers & Security (2018)

3. Bichsel, A., Hall, J., Schonning, N., Decker, J.: Configure and validate exclusions based on extension, name, or location (Oct 2018), https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-antivirus/configure-extension-file-exclusions-windows-defender-antivirus
4. Bulazel, A., Yener, B.: A survey on automated dynamic malware analysis evasion and counter-evasion: Pc, mobile, and web. In: Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium. p. 2. ACM (2017)
5. Falco, J., Lochner, M., Teumim, D.: Guidance and performance impact testing to support the use of antivirus software on scada and industrial control systems (2005)
6. Haffejee, J., Irwin, B.: Testing antivirus engines to determine their effectiveness as a security layer. In: 2014 Information Security for South Africa. pp. 1–6. IEEE (2014)
7. Heddings, L.: Antivirus slowing your pc down? maybe you should use exclusions (Nov 2010), https://www.howtogeek.com/howto/35332/antivirus-slowing-your-pc-down-maybe-you-should-use-exclusions/
8. Hsu, F.H., Wu, M.H., Tso, C.K., Hsu, C.H., Chen, C.W.: Antivirus software shield against antivirus terminators. vol. 7, pp. 1439–1447. IEEE (2012)
9. Iwer, L., Hall, J., Poggemeyer, L., Tobin, J.: Plan for hyper-v security in windows server (Mar 2018), https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/plan/plan-hyper-v-security-in-windows-server
10. Kennedy, J., Satran, M., Granito, D., Whitney, T.: Protecting anti-malware services (2018), https://docs.microsoft.com/en-us/windows/desktop/services/protecting-anti-malware-services-
11. Maxion, R.A., Townsend, T.N.: Masquerade detection using truncated command lines. In: Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on. pp. 219–228. IEEE (2002)
12. McIntosh, T.R., Jang-Jaccard, J., Watters, P.A.: Large scale behavioral analysis of ransomware attacks. In: International Conference on Neural Information Processing. pp. 217–229. Springer (2018)
13. Min, B., Varadharajan, V.: A novel malware for subversion of self-protection in anti-virus. Software: Practice and Experience **46**(3), 361–379 (2016)
14. Min, B., Varadharajan, V., Tupakula, U., Hitchens, M.: Antivirus security: naked during updates. vol. 44, pp. 1201–1222. Wiley Online Library (2014)
15. Moser, A., Kruegel, C., Kirda, E.: Limits of static analysis for malware detection. In: Computer security applications conference, 2007. ACSAC 2007. Twenty-third annual. pp. 421–430. IEEE (2007)
16. OPSWAT: Windows anti-malware market share report (2018), https://metadefender.opswat.com/reports/anti-malware-market-share\#!/?date=2018-11-26
17. Pauli, D.: Crims using anti-virus exclusion lists to send malware to where it can do most damage (Dec 2016), https://www.theregister.co.uk/2016/12/07/clever_crims_using_av_exclusion_lists_as_malware_safe_harbour
18. Perdisci, R., Lanzi, A., Lee, W.: Classification of packed executables for accurate computer virus detection. Pattern recognition letters **29**(14), 1941–1946 (2008)
19. Poulios, G., Ntantogian, C., Xenakis, C.: Ropinjector: Using return oriented programming for polymorphism and antivirus evasion. Blackhat USA (2015)
20. Quarta, D., Salvioni, F., Continella, A., Zanero, S.: Toward systematically exploring antivirus engines. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 393–403. Springer (2018)
21. Rubenking, N.: The best antivirus protection of 2018, https://au.pcmag.com/antivirus/8949/the-best-antivirus-protection
22. Sancho, D.: Microsoft virus scanning recommendations bring risks (Dec 2009), https://blog.trendmicro.com/trendlabs-security-intelligence/microsoft-virus-scanning-recommendations-bring-risks/
23. Shevchenko, A.: The evolution of self-defense technologies in malware-securelist. Securelist-Information about Viruses, Hackers and Spam. SecureList **28** (2007)
24. Swinnen, A., Mesbahi, A.: One packer to rule them all: Empirical identification, comparison and circumvention of current antivirus detection techniques. BlackHat USA (2014)
25. Watters, P.A., Watters, M.F., Ziegler, J.: Maximising eyeballs but facilitating cybercrime? ethical challenges for online advertising in new zealand. In: 2015 48th Hawaii International Conference on System Sciences. pp. 1742–1749. IEEE (2015)
26. Zhu, Z., Dumitras, T.: Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 458–472. IEEE (2018)