Measuring Malware in Online Game Markets

Esther Goldstein
Stanford University

Jackson Lallas
Stanford University

Maya Ziv
Stanford University

1 Abstract

How much malware can be found in popular gaming websites? We attempt to answer this question by sampling free Windows games from itch.io and Steam, two of the largest PC gaming marketplaces. We crawl these sites and download game binaries in order to find instances of malware. VirusTotal provides us with a starting point as to which games are infected, and we use its output to determine patterns of malicious games, efficacy of antivirus software, and develop recommendations for each store.

2 Introduction

The online PC game marketplace has been growing rapidly. The dominant platform Steam, which is operated by Valve, now has 90 million monthly users and a collection of 30,000 games [4]. Of these games, 9,700 were released in 2018 and 6,700 in 2017 [3]. In two years the size of the Steam library has more than doubled and this growth shows no signs of stopping with game development tools becoming increasingly accessible.

The recent surge in game development has allowed new stores to challenge Steam's dominance. Many developers now produce games independently and offer out of the box experiences not well suited for the expensive, burdensome process of getting on Steam. itch.io has emerged as a haven for independent and often quirky games, with over 120,000 games available [6]. Though itch.io does not publish active user data, it has a tremendous amount of goodwill among consumers and is often cited as the best alternative to Steam [6], [17].

As these platforms become more popular and face challenges vetting the thousands of games, users may face a risk of malware disguised as a game. Despite Valve's vast resources, there have been a number of high profile cases of malware on Steam. "Abstractism" ran cryptocurrency miners and "Dynostopia" installed keyloggers, webcam worms, and would corrupt all of a user's files [2], [5]. These markets are a particularly attractive target for adversaries. Within the Google Play store, malicious apps can generate hundreds of thousands per month if they can sneak on [10]. Game executables are often installed and run directly by the user, making delivery of malware relatively simple. Games can also perform behavior useful for malware writers without raising user suspicion: they commonly automatically install updates, exchange packets over the network, write to the file system, and are generally granted permissions for user convenience. There has not yet been a study on malware incidence within online PC gaming marketplaces.

Our research estimates the amount of malware present among free Windows games on Steam and itch.io. This estimate can help protect end users, evaluate antivirus software, and give direction to future work that analyzes game malware more closely. We will begin by discussing related work (Section 2) and then outlining our methodology for collecting and scanning games (Section 3). Then we discuss the results of our scans, including malware rates, malware survival times, and comparison between stores (Section 4). We provide recommendations based on these results (Section 5) and limitations of our methodology (Section 6). The final section concludes with a summary of our work and the central results (Section 7).

3 Related Work

Our study draws heavily from the previous work of Kikuchi et al. [9]. They analyzed the presence of malware on Android stores and policies used to fight malware. Game binaries were submitted to VirusTotal to determine whether or not the application was malicious. They show that malware tends to cluster along specific categories and also established malware survival times as a metric for how effectively stores protect users. Both of these metrics are integrated into our analysis, and this study also relies on VirusTotal to detect malware.

Antivirus software commonly flags innocent games as malware, leading to many false positives [16]. The false positive rate can lead to mixed results from VirusTotal, with the antivirus software not reaching consensus on the potential

malicious behavior of the game. To help detect malware in the midst of mixed results, Rahman et al. explored using store page and meta information to identify game malware. They incorporated meta information beyond the game binary to determine whether or not a game is malicious [11]. Metrics include information about the account that posted the game, reviews and how correlated those reviews are to fraudulent ones, and examination of permissions requested by applications. We propose similar identifying features of malware and use their method to help classify applications when VirusTotal provides unclear results.

There has also been robust work in the Android and Google Play stores on complex malware detection techniques. Methods include application of machine learning, analysis of runtime data such as CPU usage and network packets, binary scans using antivirus software, examination of permissions that applications request, monitoring system calls, and applying blockchain techniques [7], [8], [11], [13], [15]. This study primarily aims to characterize the current risk of malware in the PC gaming space, so we leave more thorough behavioral analysis of malicious games to future work.

4 Methodology

We crawled each store to identify all of the free games currently hosted. We then downloaded a sampling of games from each platform and uploaded the sampled binaries to VirusTotal in order to find the percentage of malware. Except for crawling the Steam store, there were no open source tools to handle each step of this process. We wrote Python scripts from scratch to automate our data collection, downloads, and uploads.

4.1 Steam

Our scan of the Steam store identified 1,192 free games. We used the open source project Steam Scraper to perform the scan [14]. Steam requires all downloads to be processed through its desktop client. This presented technical challenges that prevented us from using Headless Chrome or Firefox to download directly from the web. Instead, we produced a script that used Valve's steamcmd API to automate the downloads. Of the 1,192 games found we were able to download 1,142 and successfully uploaded 1,091 of these to VirusTotal.

4.2 itch.io

At the time of our crawling, itch.io displayed 85,649 free Windows games on its storefront. We sampled 15,000 games and 13,913 of these were successfully downloaded and uploaded to VirusTotal. First we crawled the site and created a list of all of the URLs for each game's website. To sample we used Python's random.shuffle() on the entire list and then selected the first 15,000. Next we used Headless Chrome [1] to visit each URL and attempt to download the associated game file. We looked at using Headless Firefox as an alternative, but it required us to whitelist all of the MIME types we wished to download, which was difficult to determine ahead of time since many of the games had unusual extensions. After downloading the games, we extracted any archive files in order to retrieve the binaries for upload.

4.3 VirusTotal

Once the games were downloaded and extracted into their respective folders, the next task was to find the relevant executable in each of them and upload it to VirusTotal for analysis. In order to accomplish this, we first had to detect the relevant executable in each folder. To this end we compiled a list of filetypes we'd seen and ranked them in order of priority (executables first, then .app, .bin, etc), searching through all the files in a game's extracted directory for a file of that type. See appendix A for a list of all of the different extensions we saw. We also had to filter out some common executables that came with certain types of games; for example, Unity games had a UnityCrashHandler.exe that we needed to ignore to get the actual game file. Once we'd identified the relevant file, we uploaded it to VirusTotal via their academic API, retrieving a set of scan results. Due to size restrictions, we were only able to upload files less than 200MB. Some games were simply too big for this, which accounts for the 8% of games that we weren't able to successfully scan.

5 Results

5.1 Choosing a Detection Threshold

VirusTotal does not return a ground truth label as to whether or not a certain upload is malware; instead it offers the reports of around 70 different antivirus agents analyzing that binary, as well as some sandboxed behavior analysis. For this work, we utilize the strategy developed in [9], by examining the *Positive Detection Ratio (PDR)* for each game, or the number of AV agents which flag a game as Malware over the total number of agents which were able to scan the file. In order to determine the rates of malware present in each store, we needed to determine a reasonable PDR threshold above which we consider games to be malware. Setting this threshold too high means we would miss real malware, and setting it too low would result in too many false positives. Figure 1 shows the ratio of the store that is malware for each PDR threshold.

We can see that the rates of malware even out when the threshold is in the range of 15-30%. However, deciding where to set our threshold within that range is a tricky task. We know anecdotally that it's likely that Steam has a lower MPR than itch given that binaries are scanned before uploading to Steam and anyone (including individuals) can upload to itch's storefront. Visually this would imply that the threshold

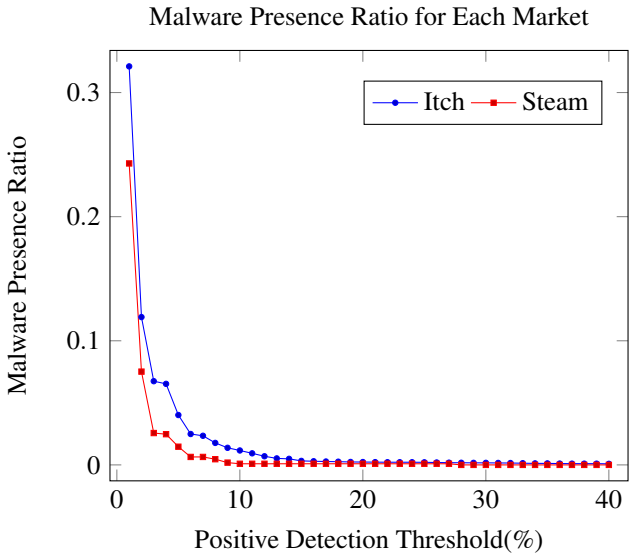

Figure 1: The ratio of each market that is malware plotted by where we set the minimum threshold for positive detection ratio.

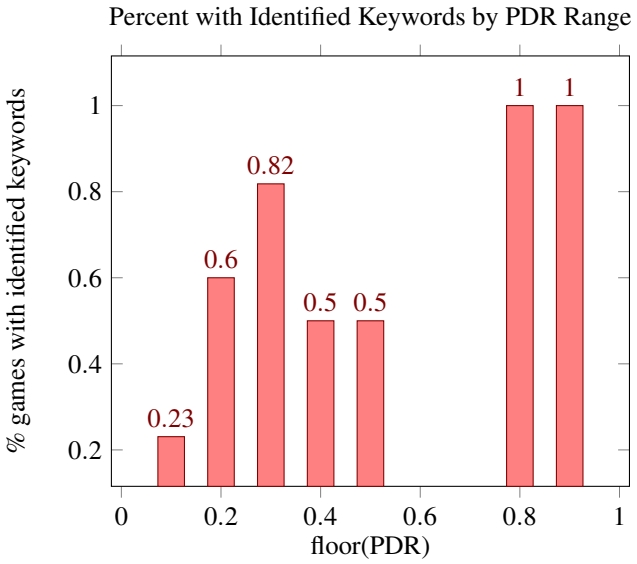

Figure 2: The percentage of games in each PDR bucket which had a keyword identified for them.

should be closer to 15%. However, we also don't want to risk including too many false positives; Kikuchi et. al's paper set their threshold to 30%. Thus, we need to approximate some kind of ground truth in order to pick a reasonable estimate.

Given the low prevalence of malware on both stores (around 20 - 40 games in the itch sample and 0 - 1 games on Steam), we had the capacity to do a closer analysis of the games that were flagged. All of the AV agents on VirusTotal gave back more than just a positive or negative flag; they also gave back a descriptive label identifying what kind of malware they thought the binary was. Generic labels included things like *"Win/malicious_confidence_60% (W)"* and *"Malicious (high Confidence),"* and more specific labels were things like *"Trojan.GenericKD.42060345"* and *"Win32.Neshta.A"* identifying specific strains of malware. Our litmus test for whether or not something was definitely malware was to examine how many of the AV agents that flagged the game agreed on what type of malware it was, especially if they all identified a particular strain. For example, the game "Jedi Vs. Sith Ultimate Notebook Edition" was flagged by 67/70 agents, making it almost certainly malware. Upon examining its labels, we see that 60% of them contain the keyword "Neshta," identifying it as the Neshta strain of malware.

Formatting across agents was not the same, so our strategy for detecting keyword agreement across them was to examine the most common substrings shared between the labels. Our strategy was as follows:

1. Perform a LCS search across each unique pair of labels, ignoring results < 4 characters long
2. Map from LCS to # of occurrences across all pairs
3. Sort by length of LCS x # of occurrences (prioritize longer and common substrings)
4. Manually examine the top 10 to identify names corresponding to types of Malware
5. If a LCS is an identifiable type of malware and is a substring of at least 20% of agents' labels (and at least 3 total), it is a valid keyword

We said that a keyword was an identifiable type of malware if it was listed in either Symantec, Microsoft, or F-Secure's online databases, chosen based off of their seeming comprehensiveness. Based on this strategy, we were able to identify keywords for 60% of games with a PDR of at least 15%. Figure 2 is the distribution of what percentage of games in each PDR category had valid keywords identified for them. Note: A game's PDR category is the floor of its actual PDR. See appendix B for a list of the identified keywords.

We can see that all the games flagged by at least 80% of agents had valid keyword matches, which implies that the metric is sound for identifying real malware. We also see that for all games identified by >= 20% of agents, at least half of them were keyword-identifiable; most of the games flagged by < 20% of agents were not keyword-identifiable, and the rest were only matched by the minimum number of agents (three). Note that there were no games in the 0.6-0.7 categories. Based on these results, we decided to set our PDR threshold to 20% for the bulk of our analysis, as this strikes the best balance between avoiding false negatives and minimizing false positives.

Using a threshold of 0.2 as the minimum PDR, we see

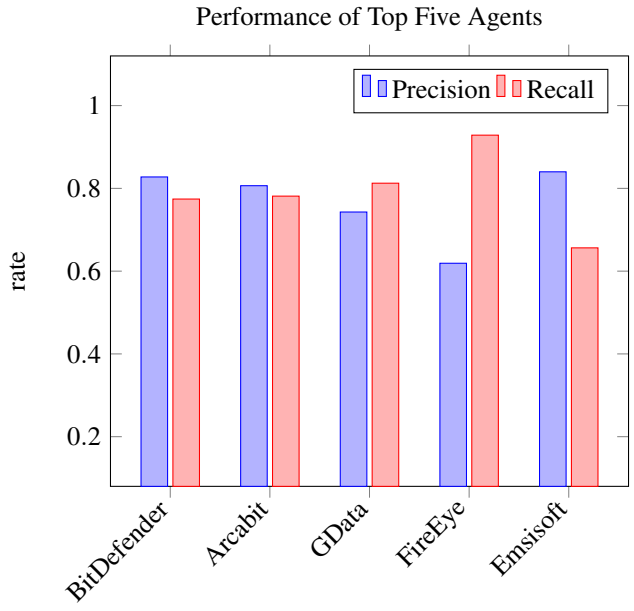

Figure 3: Precision and recall for the top five antivirus agents, calculated across both stores.

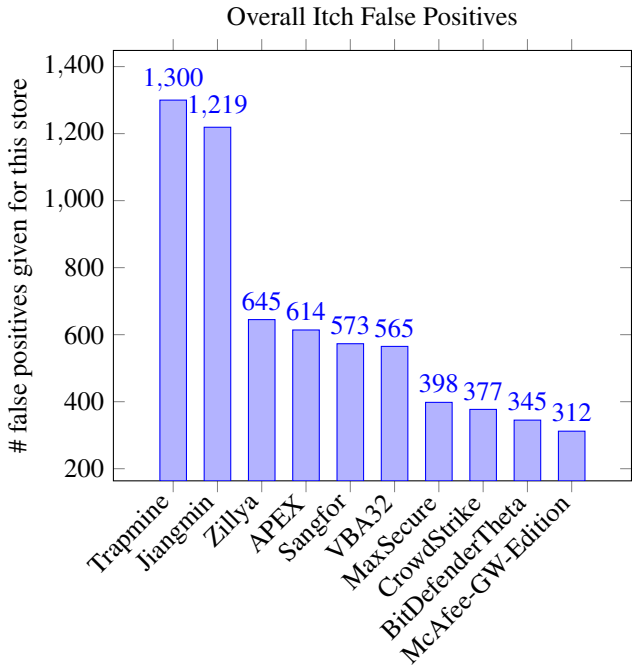

Figure 4: The number of false positives generated by the top 5 most flag-happy AV engines on itch.io data.

that the malware presence percentages of free games on each marketplace are as follows:

itch.io	0.2296%	(32 / 13,913 games)
Steam	0.0917%	(1 / 1091 games)

5.2 AntiVirus Agent Efficacies

The next piece of analysis we wanted to do was to examine the efficacy of each agent run on VirusTotal; in our preliminary research we saw a few times on Steam forums that otherwise decent antivirus software would flag games on the Steam store and prevent players from being able to run them, often generating the support advice "set your anti-virus to Game Mode or disable it before launching Steam if you are experiencing issues with your Steam games.", implying that false positives were an issue [16]. Similarly to Kiuchi et. al, we also found that most of the positive flags came from just a few incredibly sensitive agents; on Steam 43% of positives came from just three agents, and on Itch 34% of detections came from the top three agents.

We wanted to see, based on our data, which agents performed the best overall. In order to do this, we calculated the precision and recall for each engine and sorted them by the sum of their scores. The top five are listed in Figure 3.

A full table of results for each AV agent can be found in appendix C. Worth noting, Figure<x> is the table of results for some popular agents. Notice that many of them have quite low precision, resulting in a high false positive rate. This helps explain the Steam support team's advice.

Agent	Precision	Recall
Microsoft	0.17	0.84
McAfee-GW-Edition	0.07	0.81
F-Secure	0.41	0.47
Kaspersky	0.47	0.22
AVG	0.61	0.72

Figures 4 and 5 show the graphs of the highest numbers of false positives by agent in each store. We can see that the worst offenders are similar across the two marketplaces. Users of the Jiangmin and Trapmine antivirus software should note in the appendix their precision rates of 0.01 and 0.02 respectively, meaning that 99 and 98% of the games they flag as malware are false positives, rendering them nearly useless.

5.3 Patterns

We used our data to search for patterns within malicious games, such as whether they existed in specific categories, masqueraded as other popular games, or revealed their malicious nature over time.

In addition to the scanning, we sought to identify patterns among the flagged games. These patterns were all compiled from data that the end user could examine before downloading the game, in the same way Rahman et al. examined meta information [12]. Manually examining the games flagged by VirusTotal revealed many similarities on their store pages. Every file made some effort to masquerade as a legitimate game. However, there were frequent grammatical errors, poor

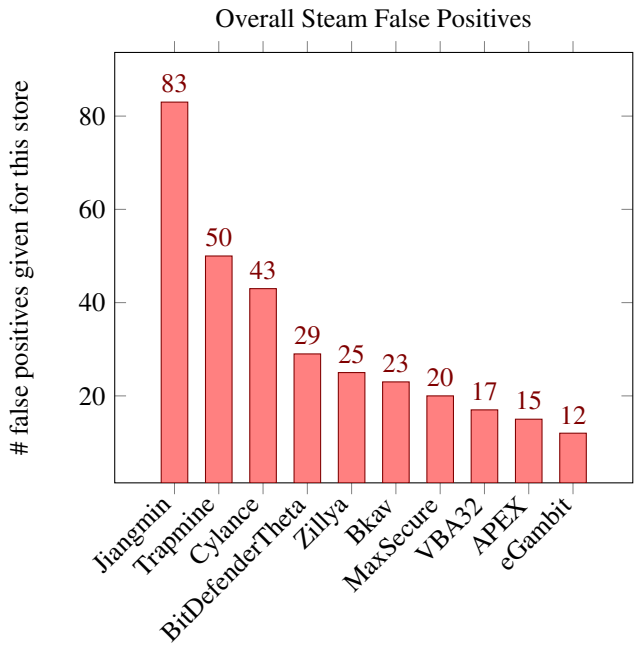

Figure 5: The number of false positives generated by the top 5 most flag-happy AV engines on Steam data.

graphic design, and short or uninformative descriptions on many pages of the flagged games. Some games were found which had pages that warned people to ignore their anti virus software or where users had commented that the game was a virus (see figures 6 and 7).

Many of the games had small files (< 5 MB) and were advertised as beta versions, proof of concepts, or single level games. Possibly such lightweight executables are designed to be the minimum necessary work to hide a malware payload within a game without raising user suspicion. The games were also often advertised as developed during game jams, events were independent developers race to create a game in a limited amount of time. This could be designed to lower user expectations about the game, as the end user does not expect a finished product. This marketing could be another way to ship a thin wrapper around malware while maintaining end user trust. It is worth noting as well that some of these games might in fact be false positives; indie games/game jam games frequently come from unverified publishers, occasionally resulting in overcautious virus flags.

Likely malware was spread across a variety of game genres. The two most popular were Action (8 games) and Platformer (5 games). Malicious applications also illegally used intellectual property to attract users. We noticed four games that did this: "Jedi vs Sith Ultimate Notebook Edition," "Puella Magi Mami Tomoe Game," "Tank City," and "King's Valley." The first two borrow characters and art from entertainment media, and the latter two are direct copies of games from older consoles like the NES. Though there were few cases of

Figure 6: In game capture of "The Curse of Mr. Hobo," which has an unpolished appearance. It had a PDR of 0.33 and was keyword-flagged as "generickd" malware.

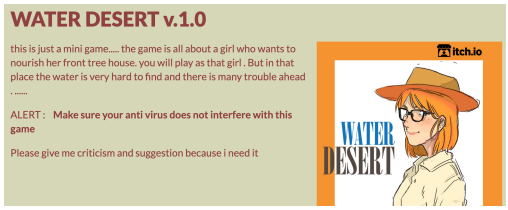

Figure 7: Screenshot of the page for "Water Desert," showcasing grammar and graphic design problems. Note the warning against Anti Virus. It had a PDR of 0.93 and was keyword flagged as "ramnit" malware.

intellectual property usage, the questionable legality of such games could send a stronger signal about malicious nature than genre does.

Survival time, the length of time that a piece of malware remains on a store, was often well over a year. One explanation could be that these games are fairly unpopular, but Itch does not release download statistics on specific games so this cannot be verified. Startlingly, the game "Jedi vs Sith Ultimate Notebook Edition" remained on the itch two years after a commenter on its store page submitted a VirusTotal report showing that the game contained malware. Though Itch does not release upload dates for games, we were able to roughly estimate the survival period of its applications that hosted time stamped content (such as a youtube video tutorial) on its store page.

The games "Forward Always Forward," "Cherchez," "Jedi vs Sith Ultimate Notebook Edition," and "thiefrl2-2016-7drl" have all been hosted on Itch for over two years. Surprisingly, the likeliest malware candidate on Steam, "Offscreen Colonies," has been on the store since 2015. The prevalence of long survival times on the Itch store indicates that Itch does not extensively scan games for malware, though their policy is not publicized. Despite the survival time of "Offscreen Colonies," the low overall rate of malware on Steam shows that the binary scanning employed has been fairly effective.

6 Recommendations

itch.io lacks any particular policy about reporting malware on its site, and each game merely has a small "Report" link at the bottom of its individual page. itch should make reporting malware easier and add malware as a reason to report a game. Platforms like itch.io should implement automated malware detection policies to reduce the probability that its users will be exposed to malware. "Jedi vs Sith Ultimate Notebook Edition," determined to be malware by 96% of engines on VirusTotal, has been on itch.io for over two years. If stores were to make the reporting process easier as well as act on these reports, the occurrences of malware on game sites could be significantly reduced.

Users may not be able to trust antivirus as the ground truth because of the false positive rate we have noted. These false positives can make antivirus cumbersome to use since users may not understand how to interpret the results. If users receive many flags on trustworthy games, they may choose to ignore their antivirus altogether. As a starting point, game stores should try to educate users on the "red flags" of malware, such as illegal use of intellectual property and counterfeit versions of games. Better user education could lower the cost of false positives and make users more likely to be wary of warnings on potentially dangerous games.

From a user interface standpoint, sites like itch.io could benefit from imposing a more consistent structure of each game's webpage. This would not only help users navigate the site more effectively, but also help the development and use of crawlers to scrape information from the site.

More work needs to be done to explore ways that end users can obtain information about the security of online stores. Both Steam and itch.io have no information about their malware policy, and we did not find any third parties auditing each marketplace for malware. Without this information, it is difficult for end users to make informed security decisions about which platform to use.

7 Limitations and Challenges

Our methodology did not get a full view of free games or the itch store. We only examined Windows games on each platform. Windows games are the most popular and supported on each store, so the bias introduced by this decision should be minimal. Our itch sample of 13,913 itch games was small relative to the 85,649 free Windows games on the store.

Our pattern analysis only examined the games flagged by VirusTotal and most of our team did not have extensive, previous exposure to the itch store. It could be the case that many of the patterns that we identified are normal for itch games, such as game jam games being common, so this information would not help the user differentiate between a safe and malicious game. The operators of itch and future work could take a larger view of the market to determine whether these features of flagged games are good indicators of malicious software.

We faced a variety of challenges creating and automating our analysis that a future study with more time available could address to build on our work. Building fault-tolerant downloaders was challenging since HeadlessChrome and Steamcmd have many scenarios where exceptions occur. This was often magnified by itch.io's overall inconsistency. Each game is hosted on its own subdomain, and even though itch provides an HTML template for these sites, game developers have freedom in how they wish to structure their sites. While some hosted their games directly on their sites, others linked to Google Drive or MediaFire, which was outside the scope of our script. During our extraction phase, we found that the downloaded games had over 50 extensions. Not all of these could be extracted using our script, so we manually extracted them.

Finally, our reliance on VirusTotal makes the results vulnerable to the same false positive problems that extensively occur with innocent games. Though we tried to minimize the chance of a false positive by relying on consensus between different antivirus agents and use of a PDR, there's still a risk that some of the games we identified were harmless. This is especially likely with "Offscreen Colonies," since it is hosted on Steam and they employ binary scanning. Future work should refine methods of classifying games as malware and directly examine the behavior of each game in a sandbox.

8 Conclusion

We found that malware was present on both Steam and itch.io, but at an order of magnitude greater rate for itch.io. We analyzed results from VirusTotal to find that many of these games were low quality, beta versions likely designed to trick the user into trusting a poor product. Although we found a relatively low percentage of malware compared to the games we scanned, we only randomly sampled 16% of itch.io. It is possible that there may be many more instances of malware that we haven't yet discovered. Our results emphasize the need for automated binary scanning and . We hope that our recommendations increase the security practices of each store and inform users about the tell-tale signs of malware.

Acknowledgements

We would like to thank Zakir Durumeric for his feedback and guidance throughout the project. Additionally, the students of CS 356 provided helpful questions and insights on our presentation.

References

[1] Headless chrome crawler. https://github.com/yujiosaka/headless-chrome-crawler.

[2] Christopher Atwood. Beware of malware on steam greenlight. GameCrate, 2015. https://www.gamecrate.com/beware-malware-steam-greenlight/11657.

[3] Jonathan Bolding. Steam now has 30 thousand games. PCGamer, 2019. https://www.pcgamer.com/steam-now-has-30000-games/.

[4] Wes Fenlon. Steam now has 90 million monthly users. PCGamer, 2019. https://www.pcgamer.com/steam-now-has-90-million-monthly-users/.

[5] Gabe Gurwin. Valve bans steam game that was installing cryptocurrency mining malware. DigitalTrends, 2018. https://www.digitaltrends.com/gaming/steam-game-allegedly-mining-cryptocurrency/.

[6] Patricia Hernandez. The game store that outshines steam by staying small and weird. The Verge, 2018. https://www.theverge.com/2018/11/29/18118217/itchio-steam-leaf-corcoran-pc-games-indie.

[7] Ali Dehghantanha Reza M. Parizi Homayoun, Sajad and Kim-Kwang Raymond Choo. A blockchain-based framework for detecting malicious mobile applications in app stores. In *arXiv:1906.04951*. arXiv, 2019.

[8] Yasir Malik Jaiswal, Mayank and Fehmi Jaafar. Android gaming malware detection using system call analysis. In *6th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, 2018.

[9] Yosuke et al. Kikuchi. Evaluating malware mitigation by android market operators. In *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*, 2016.

[10] Lily Newman. How malware keeps sneaking past google play's defenses. 2017. https://www.wired.com/story/google-play-store-malware/.

[11] et al. Peng, Hao. Using probabilistic generative models for ranking risks of android apps. In *Proceedings of the 2012 ACM conference on Computer and communications security. (ACM)*, 2012.

[12] Mizanur Rahman Bogdan Carbunar Rahman, Mahmudur and Duen Horng Chau. Fairplay: Fraud and malware detection in google play. In *Proceedings of the 2016 SIAM International Conference on Data Mining.(Society for Industrial and Applied Mathematics)*, 2016.

[13] et al. Sarma, Bhaskar Pratim. Android permissions: a perspective combining risks and benefits. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. ACM, 2012.

[14] Steam Scraper. 2017. https://github.com/prncc/steam-scraper.

[15] Asaf et al. Shabtai. "andromaly": a behavioral malware detection framework for android devices. In *Journal of Intelligent Information Systems 38.1*, pages 161–190, 2012.

[16] Steam Support. Antivirus software reports steam games are malicious. support.steampowered.com/kb_article.php?ref=4361-MVDP-3638.

[17] Robert. Zak. The best steam alternatives for pc gamers. 2019. https://www.techradar.com/news/best-steam-alternatives.

A File Extensions

.exe, .bin, .jar, .lua, .apk, .unity,
.unity3d, .app, .x86, .caproj,
.AppImage, .quest, .odp, .uproject,
.py, .quest, .sb2, .sb3, .bat, .html,
.xlsm, .ppsx, .love, .bat, .pck, .msi,
.pptx, .p8, .pde, .yyp, .gm81, .yyp,
.gmz, .dmg, .webloc, .el, .livecode,
.EXE, .aslx, .blend, .swf, .stencyl,
.z80, .gblorb, .vbs, .json, .rpgproject,
.ps1, .PNG, .dll, .iso, .mfa, .js,
.tar.bz2, .tap, .tar.bz, .rbxl, .zip,
.7z, .tar.xz, .RAR, .rar, .sb2, .pdf,
.txt, .wav, .mp4

B Identified Keywords

relevantknowledge,
trojan.genericd.32696589,
neshta, csfrsys, relevant,
jaik,
adware,
strictor,
application.relevantknowledge.gen.3,
trojan.genericd.42060345,
trojan.heur.kt.2.@j0@a8!sdjei,
trojan.genericd.32753823,
trojan.genericd.32753981,
ramnit,
adware,
miner,
gen:variant.razy.546521,
swisyn, gen:variant.razy.587815,
trojan.genericd,
gen:variant.jaik.37330,
trojan.genericd.41731057,
application.relevantknowledge.g,
ramnit,
killprocs.a,
relevant

C AntiVirus Agent Statistics

Agent	Precision	Recall
BitDefender	0.83	0.77
Arcabit	0.81	0.78
GData	0.74	0.81
FireEye	0.62	0.93
Emsisoft	0.84	0.66
Fortinet	0.8	0.62
Ad-Aware	0.85	0.53
MicroWorld-eScan	0.82	0.56
Symantec	0.54	0.81
MAX	0.76	0.59
AVG	0.61	0.72
ESET-NOD32	0.88	0.44
Alibaba	0.82	0.44
ALYac	0.79	0.47
Endgame	0.57	0.66
AegisLab	0.44	0.75
Avast	0.67	0.52
Malwarebytes	1.0	0.16
Zoner	1.0	0.16
Baidu	1.0	0.12
VIPRE	0.82	0.28
Kingsoft	1.0	0.09
eGambit	0.08	1.0
NANO-Antivirus	0.61	0.44
ViRobot	0.86	0.19
Panda	0.69	0.34
Sophos	0.62	0.41
Microsoft	0.17	0.84
Qihoo-360	0.37	0.59
TrendMicro-HouseCall	0.54	0.41
Avira	0.43	0.5
K7AntiVirus	0.55	0.38
K7GW	0.55	0.38
McAfee-GW-Edition	0.07	0.81
Invincea	0.28	0.59
F-Secure	0.41	0.47
McAfee	0.11	0.75
TrendMicro	0.56	0.28
TACHYON	0.75	0.09
Acronis	0.41	0.41
ZoneAlarm	0.58	0.22
TotalDefense	0.62	0.16

Agent	Precision	Recall
Rising	0.09	0.69
Comodo	0.28	0.47
Trapmine	0.02	0.72
Cybereason	0.19	0.53
AhnLab-V3	0.47	0.25
Paloalto	0.23	0.47
Cyren	0.41	0.28
MaxSecure	0.02	0.67
Kaspersky	0.47	0.22
APEX	0.03	0.62
Cylance	0.05	0.59
DrWeb	0.26	0.38
CrowdStrike	0.04	0.53
BitDefenderTheta	0.04	0.52
Ikarus	0.2	0.34
Webroot	0.21	0.32
Sangfor	0.02	0.47
CAT-QuickHeal	0.18	0.26
F-Prot	0.31	0.12
VBA32	0.02	0.41
Yandex	0.14	0.25
SentinelOne	0.13	0.25
Tencent	0.14	0.16
Zillya	0.01	0.28
Jiangmin	0.01	0.28
Antiy-AVL	0.03	0.25
ClamAV	0.1	0.16
Bkav	0.02	0.22
CMC	0.12	0.09