I Know Your Antivirus: Revealing Client-Side Antivirus via Browser Clue

Hyeong-Seok Jang[1][0009-0000-8167-0951], Se-Han Lee[1, 2][0009-0009-5919-4157] and Ki-Woong Park[3, †] [0000-0002-3377-223X]

[1] SysCore Lab., Sejong University, Seoul 05006, Republic of Korea
[2] Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea
[3, †] Dept. of Computer and Information Security, Sejong University, Seoul 05006, Republic of Korea
rotiple320@gmail.com, sehanlee141@gmail.com, woongbak@sejong.ac.kr(Corresponding Author)

Abstract. As a cyberattacks utilizing cloud infrastructure and 0-day vulnerabilities become increasingly sophisticated, advanced persistent threats (APTs) are evolving. One of the primary goals of an APT is system detection, and an attacker's ability to detect the security software on a system is a critical factor in the success of the attack. In this work, we developed and experimented with a testbed to detect locally installed antivirus in a web browser environment. This work analyzes how antivirus information is exposed in the web browser and evaluates if it can be used for detection. The testbed consists of *Orchestrator*, *System Snapshot Repository*, *Client X*, *Profile Server*, and *Collector*, and is designed to test antivirus detection scenarios across a variety of antiviruses and web browser environments. We detected specific antiviruses using methods such as DOM monitoring, signature detection, and phishing page detection. This work detects vulnerabilities in antivirus detection in web browser environments and contributes to future security improvements.

Keywords: Antivirus Detection, Profiling, Testbed.

1 Introduction

As cyberattacks become more complex and technologies advance [1, 2], 0-day vulnerability attacks targeting cloud infrastructure are increasing [3], and new attack methods utilizing artificial intelligence (AI) technology are also emerging [4]. This has led to the emergence of advanced attack vectors such as advanced persistent threats (APTs), which can detect antiviruses and design attacks that bypass them [5].

In this work, we built a testbed specialized in antivirus (AV) detection in a web browser environment and analyzed security vulnerabilities due to exposure of antivirus information. This work investigates how antivirus information can be exposed in client-

side web browsers and proposes a methodology to evaluate antivirus detection performance.

2 Implementation of Testbed

We built a testbed where we can test various antiviruses and analyze their operation.

The constructed testbed consists of *Orchestrator*, *System Snapshot Repository*, *Client X*, *Profile Server*, and *Collector*. These components are used to evaluate antivirus detection performance and vulnerabilities in the web browser environment. An overview of the structure and operation flow for the testbed is shown in Figure 1.

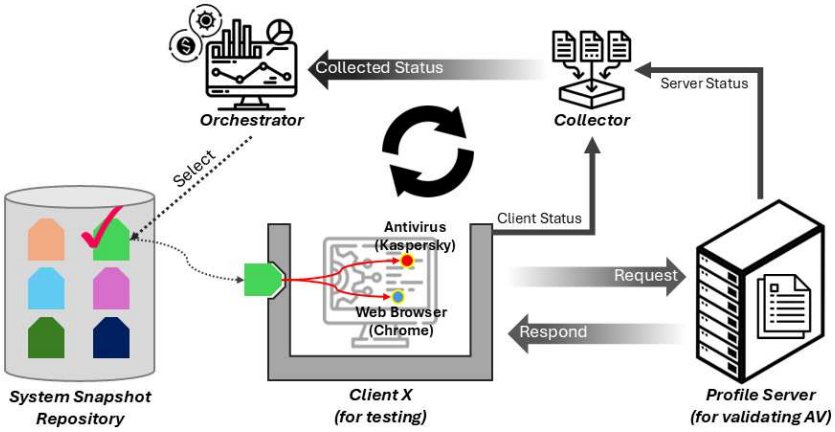

Fig. 1. An overview of a testbed architecture for browser-based antivirus detection.

The *Orchestrator* is a central control system that manages system snapshots for *Client X*, collaborates with the *Collector* to analyze collected status data, and oversees the entire experiment. The *System Snapshot Repository* stores snapshots with different antivirus and web browsers, providing suitable testing environments. The *Client X* simulates a web browser and antivirus environment to elicit antivirus responses, while the *Profile Server* provides test scenarios, and the *Collector* gathers and analyzes status data in real-time. The *Orchestrator* analyzes the collected data to assess antivirus detection performance and identify vulnerabilities. This flexible testbed supports various antivirus and web browser configurations, enabling automated, repetitive tests for detailed performance analysis.

3 A Methodology for Antivirus Profiling

In this section, we propose a methodology for profiling antiviruses in a web browser environment by analyzing their operation under different conditions. In this work, we examine antivirus responses to normal and malicious web pages to infer key features and establish a hypothesis about the conditions that trigger detection.

The hypothesis is tested by generating scenarios on the *Profile Server* and setting up different antivirus and web browser environments on the testbed. The results, analyzed through the *Collector*, help evaluate antivirus responses and identify effective detection conditions. These findings can inform new APT attack scenarios, offensive security research, and antivirus improvements.

4 Experiments

This section describes the results of experiments aimed at detecting antivirus operations in a web browser environment and evaluating the ability to identify specific antiviruses.

In our work, the three experiments were proceeded:

Experiment 1: Detection of antivirus operation through changes in the Document Object Model (DOM) structure [6]. Some antiviruses modify the DOM to control security, allowing their detection.

Experiment 2: Loading a page with the Eicar [7] test signature to check if the antivirus detects and blocks it. Most antiviruses blocked the signature, and differences in blocking messages or methods helped infer the antivirus.

Experiment 3: Loading a phishing page [8] to test antivirus detection. While most antiviruses blocked the page, many did not generate detection notifications, highlighting a potential risk in APT attacks.

In this work, three experiments were conducted targeting the Chrome browser, and the results for various antivirus detections are shown in Table 1.

Table 1. A test results for three experiments from Chrome web browser.

Type of AV	Exp. 1	Exp. 2	Exp. 3	Results
Kaspersky Premium	O	O	O	Detect
Avast Free Antivirus	X	▲	X	Partial Detect
AVG Internet Security	X	▲	▲	Partial Detect
ESET Premium	X	▲	O	Detect
McAfee Total Protection	X	X	X	Fail
Avira Free Antivirus	X	X	X	Fail
Bitdefender Total Security	X	O	O	Detect
360 Total Security	X	X	X	Fail
F-Secure Internet Security	X	X	X	Fail
K7 Total Security	X	X	O	Detect

5 Conclusion

In this work, we propose a methodology to identify antiviruses by leveraging their detection and blocking mechanisms in a web browser environment using a testbed. This demonstrates the effectiveness of detecting antiviruses in such environments and confirms the testbed's usefulness for analyzing antivirus performance and evaluating security vulnerabilities. In the future, we plan to expand the testbed to detect other security software in web browser environments and to test its applicability in mobile and cloud systems.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT (Project No. 2022-11220701, 50%; RS-2022-00165794, 20%), and the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (Project No. RS-2023-00208460, 30%)

References

1. Mallick, M. A. I., Nath, R.: Navigating the Cyber security Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments. World Scientific News 190(1), 1–69 (2024).
2. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., Akin, E.: A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. Electronics 12(6), 1333 (2023).
3. Swathy, A. M., Padmavathi, G.: Zero-Day Attack Path Identification using Probabilistic and Graph Approach based Back Propagation Neural Network in Cloud. Mathematical Statistician and Engineering Applications 71(3s2), 1091–1106 (2022).
4. Guembe, B., Azeta, A., Misra, S., Osamor, V. C., Fernandez-Sanz, L., Pospelova, V.: The Emerging Threat of Ai-driven Cyber Attacks: A Review. Applied Artificial Intelligence 36(1), 2037254 (2022).
5. Sharma, A., Gupta, B. B., Singh, A. K., Saraswat, V. K.: Orchestration of APT malware evasive manoeuvers employed for eluding anti-virus and sandbox defense. Computers & Security 115, 102627 (2022).
6. MutationObserver. https://developer.mozilla.org/ko/docs/Web/API/MutationObserver (accessed Sep. 2024).
7. EICAR. https://www.eicar.org/ (accessed Sep. 2024).
8. AMTSO Phishing Test Page. https://www.amtso.org/feature-settings-check-phishing-page/ (accessed Sep. 2024).