REPHRAIN

Protecting citizens online

A Privacy Testbed for IT Professionals: Use Cases and Design Considerations

Joseph Gardiner, Bristol Cyber Security Group, University of Bristol

Mohammad Tahaei, Bristol Cyber Security Group, University of Bristol

Jacob Halsey, Bristol Cyber Security Group, University of Bristol

Tariq Elahi, School of Informatics, University of Edinburgh

Awais Rashid, Bristol Cyber Security Group, University of Bristol

September 2021

UK Research and Innovation · University of Bristol · UCL · The University of Edinburgh · King's College London · University of Bath

A Privacy Testbed for IT Professionals: Use Cases and Design Considerations

Joseph Gardiner[1], Mohammad Tahaei[1], Jacob Halsey[1], Tariq Elahi[2], and Awais Rashid[1]

[1]Bristol Cyber Security Group, University of Bristol
[2]School of Informatics, University of Edinburgh
{joe.gardiner,mohammad.tahaei,vw18148,awais.rashid}@bristol.ac.uk
t.elahi@ed.ac.uk

Abstract

We propose a testbed to assist IT professionals[1] in evaluating privacy properties of software systems. The goal of the testbed, currently under construction, is to help IT professionals systematically evaluate and understand the privacy behavior of applications. We first provide three use cases to support developers and privacy engineers, and then describe key design considerations for the testbed.

1 Motivation

In the past 10 years, within the research community there has been a realization that IT professionals, much like end-users, need support in performing privacy and security tasks [1, 8, 13, 15]. Research studies have covered a wide range of IT professionals, including developers [8], system administrators [11], reverse engineers [18, 19], and privacy engineers [14].

Developers, as one of the main players in the software ecosystem, find it challenging to accomplish privacy and security tasks, partially because of usability issues. Poor usability of security libraries may lead to developers making critical mistakes, e.g. when trying to establish a secure connection that leaves their applications open to man-in-the-middle attacks [5, 7]. System administrators, professionals in charge of ensuring that systems run securely, also face difficulties in keeping up with security updates, deciding when to update, and understanding the consequences of those updates [10, 11, 17]. Similarly, when it comes to privacy, developers may not always be aware of what data is being collected through their applications by third-party libraries [3, 16]. System administrators face similar challenges, whether to establish if applications will violate users' privacy or to comply with requirements from regulations such as the EU's General Data Protection Regulation (GDPR).

In this paper, we focus on providing support for systematically evaluating the privacy properties of third party applications and libraries—research shows that professionals find these types of privacy tasks hard [2, 14, 16].

While it would be great if professionals were able to identify the privacy consequences of their choices and decisions before rolling out their products, evidence shows that it can be hard to be aware of all the privacy nuances of a software system. In particular, in the modern convoluted software development ecosystem, IT professionals often make use of third-party tools, libraries, and components, adding many layers of complexity to their systems.

Responses from academia have been in the form of suggestions for better design of tools that can identify privacy issues [6, 12]. Therefore, we propose a testbed that can assist IT professional to understand the privacy behavior of their applications in a controlled environment. In this short paper, we describe three example use cases for such a testbed, and provide an overview of the key functionality being developed as part of its ongoing implementation.

2 Use Cases

Our proposed testbed can assist software developers, system administrators, and privacy professionals, to run a large scale analysis without the need to deploy any infrastructure or have access to several (potentially costly) target devices. They will be able to instantiate multiple virtual devices with various versions of operating systems to facilitate executing privacy-related analyses. We outline three sample use cases here to illustrate the value in our testbed.

[1]Includes anyone who builds and maintains software systems such as software developers, system administrators, privacy engineers, and testers.

USENIX Symposium on Usable Privacy and Security (SOUPS) 2021.
August 8–10, 2021, Virtual Conference.

Use Case 1. Developer Alpha wants to test their app, that depends on multiple third-party libraries for functionality, to check if it collects unnecessary data from users. Using our testbed they can run their app on multiple virtual instances of Android and iOS devices and get reports of the transmitted network data between the app and destinations (such as servers and service platforms) on the Internet. The developer would only need to drop their app file into our user interface and follow a wizard-based tool. Our testbed runs multiple privacy tests against the file and produces a report in a comma-separated format, which the developer can download for their own analysis. To assist the developer further, the testbed provides an abstraction level where the raw data is mapped onto a privacy-evaluation framework (e.g., Privacy by Design [9] and LINDDUN [4]) to help them understand the consequences of the collected privacy sensitive data. For better coverage, our testbed will be able to run multiple analyses at once (e.g., Exodus[2] and LibRadar[3]) to facilitate finding privacy issues using multiple tools.

Use Case 2. Developer Beta wants to measure the resilience of their privacy-preserving peer-to-peer file sharing application to attacks, such as Sybil attacks or network partitioning. They deploy a large number of instances of their application on a virtual network on the testbed, designating a subset of that network to become "malicious" and launch the attacks. Depending on the attack scenario, the testbed can measure the impact on application performance whilst under attack, measure if a subset of compromised nodes can de-anoymize users, and other security, privacy, and performance metrics.

Use Case 3. Privacy engineer Gamma wants to learn about and test out modern privacy-enhancing technologies (e.g., homomorphic encryption, secure multi-party computation, and differential privacy which can be hard to understand and use [2]). They can use the testbed to run experiments on these technologies before deploying them in a final product. Such technologies can be intimidating, hard to get working, and sensitive to minor errors. Therefore, our testbed would make these PETs more approachable and accessible for those who do not need to understand the technical details and only need assurance about the properties they preserve when deploying them within their applications.

3 Testbed Design

In order to accommodate the use cases described above, these are three key functionalities of the testbed under development:

1. **Deployment** - The testbed should allow for the easy deployment of services and hosts, potentially numbering in the thousands. This should include support for traditional hosts and virtualized smartphone environments.
2. **Orchestration** - Once machines are launched, the user should be able to automate application functions in order to test at scale without manual intervention. This can include automated navigation within smartphone applications, replaying of network traffic from previous captures or simulated users of a chat application.
3. **Data Logging** - It is critical that the testbed features sufficient logging capabilities in order to allow for data analysis. As well as the obvious data such as network traffic, this can also include live memory captures from virtual hosts and automated screen captures of administrator and user screens.

The aim of the testbed is that the technology stack that supports these three areas of functionality will be application agnostic—i.e., no matter what application is being tested, the system can facilitate deploying the virtual hosts and servers, support orchestration through scripting and capture data, with automated processing of the data using existing toolsets to identify information leakage.

A further key aspect is extensibility. For an individual deployment of the testbed architecture, computation resources will limit the number of virtual hosts that can be deployed. To enable larger scale experimentation, the testbed will be extensible by design, and support multiple instances of the testbed to be connected to provide much greater virtualization capacity. This does come with additional challenges, including much more complex orchestration and bandwidth limitation on the network links between testbed instances. These are issues that we are exploring as we develop the initial version of the testbed.

4 Conclusion

In this paper, we presented a testbed design and sample use cases to assist IT professionals in evaluating privacy behavior of applications. We are currently in the prototyping stage of building a working testbed with support for machine deployment, networking provided by software-defined networking and the ability to capture network traffic originating from applications, and are in the process of finalizing the technology stacks and analysis tools which should be included. Examples of test applications that have been deployed are the Swiss Covid-19 track and trace application, and a deployment of the Signal framework. We are keen to hear opinions and thoughts on how this platform can be better suited to the needs and requirements of IT professionals – and also researchers.

Acknowledgments. This work is in support of the PETS Testbed to be developed within the National Research Centre on Privacy, Harm Reduction and Adversarial Influence Online (REPHRAIN). EP/V011189/1.

[2] https://reports.exodus-privacy.eu.org/en/
[3] https://github.com/pkumza/LibRadar

References

[1] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. You are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users. In *Cybersecurity Development (SecDev), IEEE*, 2016.

[2] Nitin Agrawal, Reuben Binns, Max Van Kleek, Kim Laine, and Nigel Shadbolt. Exploring Design and Governance Challenges in the Development of Privacy-Preserving Computation. In *Proc. of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, 2021.

[3] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason I Hong, and Lorrie Cranor. The privacy and security behaviors of smartphone app developers. In *Workshop on Usable Security (USEC'14)*. Internet Society, 2014.

[4] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1), March 2011.

[5] Manuel Egele, David Brumley, Yanick Fratantonio, and Christopher Kruegel. An Empirical Study of Cryptographic Misuse in Android Applications. In *Proc. of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13. ACM, 2013.

[6] Serge Egelman. Taking Responsibility for Someone Else' Code: Studying the Privacy Behaviors of Mobile Apps at Scale. In *2020 USENIX Conference on Privacy Engineering Practice and Respect (PEPR 20)*. USENIX Association, October 2020.

[7] Sascha Fahl, Marian Harbach, Henning Perl, Markus Koetter, and Matthew Smith. Rethinking SSL Development in an Appified World. In *Proc. of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13. Association for Computing Machinery, 2013.

[8] Matthew Green and Matthew Smith. Developers Are Not the Enemy!: The Need for Usable Security APIs. *IEEE Security and Privacy*, 14(5), September 2016.

[9] Jaap-Henk Hoepman. *Privacy Design Strategies (The Little Blue Book)*. Radboud University, 2019.

[10] Adam Jenkins, Pieris Kalligeros, Kami Vaniea, and Maria K. Wolters. "Anyone Else Seeing this Error?": Community, System Administrators, and Patch Information. In *2020 IEEE European Symposium on Security and Privacy (EuroS P)*, 2020.

[11] Frank Li, Lisa Rogers, Arunesh Mathur, Nathan Malkin, and Marshini Chetty. Keepers of the Machines: Examining How System Administrators Manage Software Updates For Multiple Machines. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. USENIX Association, August 2019.

[12] Tianshi Li, Yuvraj Agarwal, and Jason I. Hong. Coconut: An IDE Plugin for Developing Privacy-Friendly Apps. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4), December 2018.

[13] Marten Oltrogge, Nicolas Huaman, Sabrina Amft, Yasemin Acar, Michael Backes, and Sascha Fahl. Why Eve and Mallory Still Love Android: Revisiting TLS (In)Security in Android Applications. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, August 2021.

[14] Awanthika Senarath, Marthie Grobler, and Nalin Asanka Gamagedara Arachchilage. Will they use it or not? investigating software developers' intention to follow privacy engineering methodologies. *ACM Transactions on Privacy and Security*, 22(4), November 2019.

[15] Mohammad Tahaei and Kami Vaniea. A Survey on Developer-Centred Security. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, June 2019.

[16] Mohammad Tahaei, Kami Vaniea, and Naomi Saphra. Understanding Privacy-Related Questions on Stack Overflow. In *Proc. of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, New York, NY, USA, 2020. Association for Computing Machinery.

[17] Christian Tiefenau, Maximilian Häring, Katharina Krombholz, and Emanuel von Zezschwitz. Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. USENIX Association, August 2020.

[18] Daniel Votipka, Mary Nicole Punzalan, Seth M Rabin, Yla Tausczik, and Michelle L Mazurek. An Investigation of Online Reverse Engineering Community Discussions in the Context of Ghidra. In *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2021.

[19] Khaled Yakdan, Sergej Dechand, Elmar Gerhards-Padilla, and Matthew Smith. Helping Johnny to Analyze Malware: A Usability-Optimized Decompiler and Malware Analysis User Study. In *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.