REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 20-11-2015	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 24-Sep-2012 - 23-Sep-2014

4. TITLE AND SUBTITLE Final Report: Gyrus: Preventing Sensitive Information and Malicious Traffic from Leaving Computers	5a. CONTRACT NUMBER
	5b. GRANT NUMBER W911NF-12-C-0053
	5c. PROGRAM ELEMENT NUMBER 606055
6. AUTHORS Brendan Dolan-Gavitt, Wenke Lee	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Security Axioms, Inc. 3223 Windsor Lake Drive Atlanta, GA 30319 -2374	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) 61061-CS-SB2.1

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

In this project, we have studied the problem of data protection in the cloud and mobile computing environments. The goal is to provide security solutions that are attack agnostic, general, and transparent. We have successfully developed CloudCapsule for desktops and mobile devices, as well as security overlay apps for both iOS and Android. We also have started our commercialization efforts, and in particular, ready to release these products.

15. SUBJECT TERMS

cloud computing, mobile computing, security, confidentiality, integrity, virtual machines, sandbox, snapshot-and-restore, UI automation, overlay

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Wenke Lee
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 404-385-2879

Report Title

Final Report: Gyrus: Preventing Sensitive Information and Malicious Traffic from Leaving Computers

ABSTRACT

In this project, we have studied the problem of data protection in the cloud and mobile computing environments. The goal is to provide security solutions that are attack agnostic, general, and transparent. We have successfully developed CloudCapsule for desktops and mobile devices, as well as security overlay apps for both iOS and Android. We also have started our commercialization efforts, and in particular, ready to release these products.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Post Doctorates

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Faculty Supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:......	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:......	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):......	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:......	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:......	0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See Attachment

Technology Transfer

Gyrus SBIR Phase II Project Final Report
(August 24 2012 through September 30 2014)

November 16, 2014

Brendan Dolan-Gavitt (PI)
Wenke Lee (CEO)

Security Axioms, Inc.
75 Fifth Street, NW, Suite 246A
Atlanta, GA 30308

info@security-axioms.com

Foreword

In this report, we first describe the problem studied in the project, namely, data protection in the cloud and mobile computing environments. We then provide a summary of the must important results in technical developments and commercialization efforts. We also describe the details of our technical and commercialization work.

Table of Contents

Problem Statement

As computing devices (e.g. laptops, phones, tablets) become ubiquitous and applications are used in every aspect of our daily lives, data security and privacy have also become the major concern. When a user runs an application, the primary concern is that the application should not violate the confidentiality and integrity of the users data. For example, a malicious input method that logs every keystroke violates the confidentiality of the user's credentials, e.g., his bank account and password. Similarly, a compromised bank application that steals money by forging a transaction compromises the integrity of user input data. When a user's data is stored in the Cloud, the primary concern is data privacy (or, confidentiality). For example, the user's email contents should not be exposed to an unintended party.

It is tempting to develop defenses based on analysis of (specific) attack methods or application behaviors. In fact, we have been following this path to develop misuse detection approaches that catch attacks by matching predefined malicious patterns, and anomaly detection approaches that identify anything that cannot be the result of correct execution for the given input or execution environment. It has been shown over time that such defenses are consistently inadequate: misuse detection generally cannot detect new attacks, while anomaly detection are known to suffer from false positives as well as mimicry attacks. These shortcomings are *fundamental* because attackers have nearly unlimited number of ways to invent the next, new attack methods, and for many applications there is always the possibility of new, previously unknown behaviors that are not induced by attacks.

The goal of our project is to develop fundamental, new approaches to protect data integrity and confidentiality, with these high-level objectives:

- *Attack agnostic*. That is, our defense should remain effective regardless of how an attack is implemented. Rather than relying on malware analysis and pattern matching to prevent or detect infections, it is more important to focus on ways to prevent any malware from accessing the data that needs protection. This suggests that we should study how data flows through operating systems and applications.
- *General*. That is, the same mechanisms can be applied to as wide a variety of

applications and computing platforms as possible. Rather than relying on analyzing the internals of each application and operating system, e.g., via reverse engineering, which is extremely labor-intensive and not always practical, it is more important to focus on identifying and protecting the common data processing steps. Obviously, there are applications that do require a deep behavior understanding and modeling (and thus reverse engineering in the first place), but as we will discuss in this proposal, many applications share characteristics that enable an effective, common protection mechanism.

- *Transparent*. That is, the users should not need to interact with an application differently when the protection mechanism is applied. This suggests that the application workflow and GUI should remain intact when security protection is activated.

In order to be attack agnostic, our solutions need to isolate user and application data from possible attacks. Towards this end, we developed a *capsule* technology based on virtual machine snapshot-and-restore.

In order to be general and transparent, our solutions need to be based on the common characteristics of data processing in applications and systems. We observe that the applications and systems that we use most often, particularly word-processing and messaging applications and cloud servers, only need meta-data to provide the desired functionalities. For example, a messaging application (e.g., Gmail) only needs to know the recipient(s) but not the exact message contents. That is, in many cases, the real data contents, which are most often the target of integrity and confidentiality attacks, actually need not be exposed to applications and servers. Another observation is that these applications tend to have straightforward data processing logic that often involves reformatting the input with well-defined meta-data and formats to produce the output, which is often displayed to the user (e.g., in a message window). Based on these observations, we developed a *security overlay* based on UI monitoring technology.

In summary, in this project we studied the problem of providing data protection in the cloud and mobile computing environments. Our solutions need to be attack agnostic so that they can mitigate most attacks, general so that they can be applied to many applications, and transparent so that users do not have to change their workflows. In order to achieve these goals, we developed a new capsule technology based on virtual machine snapshot-and-restore techniques and an innovative security overlay technology based on UI monitoring techniques.

Summary of Results

Technical Developments

- We have implemented a working system of the desktop/laptop version of Gyrus called CloudCapsule. This system enables secure access to data stored in a remote server (e.g., the Cloud) by using virtual machine technologies to create an isolated environment on a computer so that data resides only in the capsule and is removed afterwards. In particular, we use a novel VM snapshot-and-restore technique to undo any data theft effort by malware.
- We have also implemented a mobile version of CloudCapsule using sandboxing architectures on mobile devices. It is a single app that includes all necessary functionalities so that data only resides within the sandbox of CloudCapsule and is removed when CloudCapsule is closed.
- We have also created another app to provide a general end-to-end data protection framework to any messaging app such as WhatsApp and Gmail. The main idea is to have a *security overlay* app as a proxy between a user and a target app to automatically provide data protection. In particular, the security overlay provides a transparent window over a target app's GUI window, and intercepts user input and performs security protection (e.g., encryption) and likewise intercepts app display output and removes the security protection (e.g., decryption).

Our product vision is that a user can use the desktop/laptop version of CloudCapsule to create and edit documents and the mobile version for viewing and light editing. In addition, users can use security overlay to securely communicate using existing message apps and services such as WhatsApp. We have submitted the mobile apps to Apple Store and Google Play for review, received approvals. We are conducting more testing and improvements before we release them to the market.

Commercialization efforts

- We have created YouTube demos of both the desktop and mobile versions of CloudCapsule and have been sharing them with interested parties. We have also given several in-person demos and held several meetings with potential customers and investors. For example, we met an executive team at Wipro, Grant Wagner at NSA, Kevin Borders (formerly at NSA), John Marshall at AirWatch, Glenn McGonnigle at TechOperators, and Sudip Chakrabarti at Andreessen Horowitz. Wenke Lee had participated in the Value Creation Workshop and we presented at the DoD SBIR Workshops in 2013 and 2014
- We have also demoed and discussed the security overlay technology to a number of VCs and serial entrepreneurs, including Dr. Paul Judge, Glenn McGonnigle, as well as an executive team at Samsung.
- We are continuing to improve and refine our technologies, and have continued dialogs with potential investors and potential partners.

We are ready to release our apps through several big social groups, including students of the on-line Masters in Computer Science program at Georgia Tech. We will initially focus on user studies and feedbacks, including the wish list of killer apps.

Phase II Technical Work and Results

CloudCapsule

Consider the typical workflow when a user needs to access data that is stored in a cloud server:

1. user is authenticated to the Cloud and starts a secure session;
2. downloads a file from his Cloud storage to his computer;
3. works on the file using an application(s);
4. uploads the file back to the Cloud and concludes the session.

The security goals even with malware on user's computer are:

1. authentication – only the authorized user can download and upload files from/to the Cloud;
2. confidentiality – only the authorized user can obtain the file contents;
3. integrity – only the authorized user can upload changes back to the Cloud;
4. in addition, usability of the computer should not be degraded noticeably because of security checks.

In Phase I of the SBIR project, we had already dealt with using Gyrus to capture user input to verify that a program processes the user input as intended. The main idea is to user virtual machine monitoring techniques to capture user input, and check program output to ensure that the output data is the proper processing of the input. We call this the "what you see is what you send" (WYSIWYS) policy [1]. To ensure that the real the user, not the malware, is using his password to log in, we use Gyrus to make sure that the password is entered by the user and (its hash) is sent to the cloud server.

The main challenge in protecting confidentiality and integrity when data is downloaded from the cloud and is being used by applications on the user's computer is that the user's computer may have been infected by malware and the applications can also be compromised. More specifically, in our threat model, a malware in the user computer may attempt to read or modify the data, e.g., by accessing the files that contain such data.

Our solution, CloudCapsule, is based on virtual-machine snapshot-and-restore [2]:

- Complete snapshot of a user VM at the beginning of a secure session;
- Restricted communications during session (e.g., only allow Internet connections to approved sites);
- Compete restore of the VM at the end of the secure session
  - Any read by a malware during the session is completely undone.

Figure 1 illustrates the process of using snapshot-and-restore in CloudCapsule. We assume that the virtual machine monitor is the only trusted computing base, that is, the operating system and applications may be compromised. By allowing only connections to the approved sites and using Gyrus to verify that data to these sites are authored or intended by the user, we can ensure that malware cannot steal data and upload it to a "drop site" and the malware cannot temper with data that the user intends for a legitimate site. In addition, if the malware steals data and writes to a file and hopes to send it out later, the file will be wiped out with a VM restore. That is, since the VM snapshot took place before data is downloaded onto the local computer, and the restore complete restores the computer to the state at the time of snapshot, the net effect is equivalent to nothing has happened on the local computer, including any malware actions.

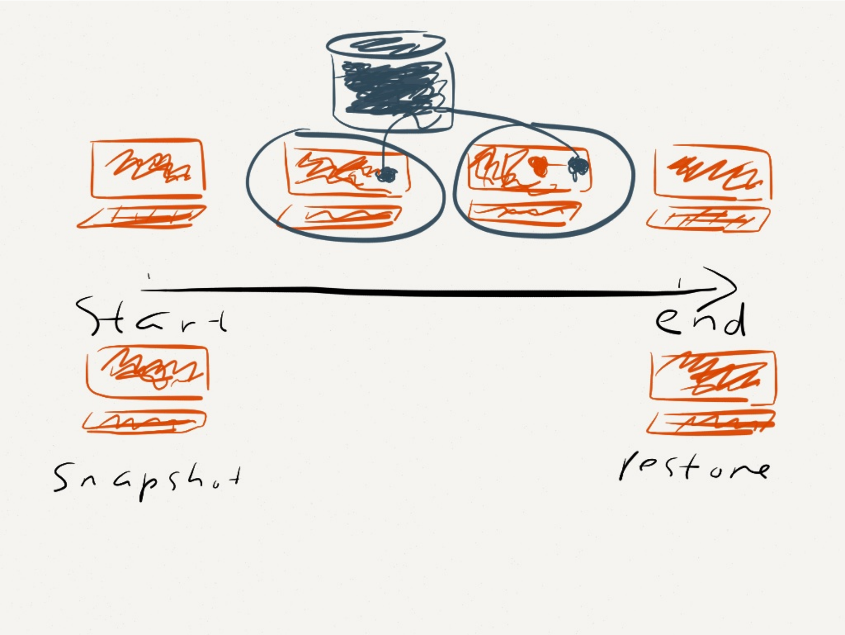


Figure 1 Snapshot-and-Store in CloudCapsule

We have implemented CloudCapsule using Linux KVM as the virtual machine monitor. We have also implemented interfaces for cloud services such as Dropbox. We have tested CloudCapsule with both normal and attack scenarios and conducted user studies. The results show that CloudCapsule provides complete protection against malware, is easy to use, and has low performance overhead.

Mobile Version of CloudCapsule

Since users often need to access data stored in the cloud using their mobile devices, as illustrated in Figure 2, we also developed a mobile version of CloudCapsule on both iOS and Android. These implementations of mobile CloudCapsule do not use virtualization technologies. Rather, we assume that the underlying operating system is trusted and in particular, the application sandboxing architecture can properly protect application data from other apps. CloudCapsule in these implementations is a single app that includes all necessary functionalities, e.g., connection to cloud servers, file transfer and encryption/decryption, and read (and write) functions. That is, data only resides within the sandbox of CloudCapsule, and only exists on the user's device when CloudCapsule is in use. That is, data files and memory contents are removed when CloudCapsule is closed.

The features of our mobile CloudCapsule include authentication of user to cloud storage server (such as Amazon and Dropbox), creation of capsule on server, download of data to mobile device for read/viewing of txt, doc, and pdf docs inside the CloudCapsule app, as well as web browsing within the app. In addition, we implemented a "safe" browsing feature that provides location and device anonymity.

Although the mobile CloudCapsule apps only provide read/viewing features, in principle, we can include editing features if there are open-source document editing tools that we can incorporate into our mobile CloudCapsule apps. However, even with just the read/viewing features, the mobile apps still enable a typical workflow where document editing is mostly done on desktop and the data is accessible/readable from everywhere, including mobile devices, securely.

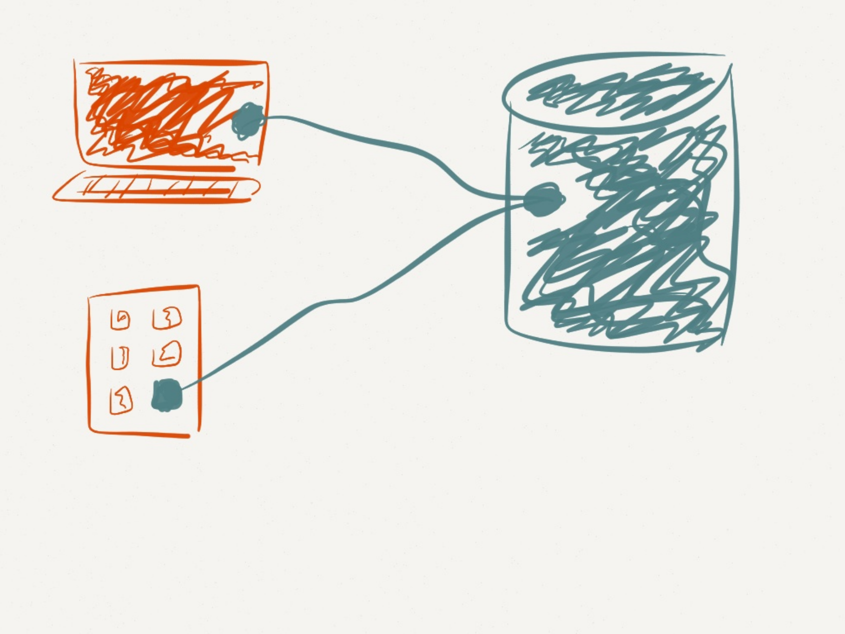


Figure 2 Mobile CloudCapsule

Security Overlay

In order to achieve the goals of general and transparent data protection, our solution needs to be based on how users form the expectation of an app's behavior, and the common characteristics of how a benign app accesses and processes private data. We observe that most apps have only very brief explanations of their features, and yet because of their intuitive user interfaces (UIs), users would know what to expect by just looking at the UIs. This suggests that semantic analysis of an app's UI should allow us to automatically deduce the user expectation of what data an app should access and how that data should be processed. Furthermore, we note that for most apps access to and processing of user data are initiated by the user through the UI. In most cases, the accessed data undergoes very little or simple processing, and often only the UI element(s) that displays data to users requires plaintext. For example, in order to function properly as a messaging application, WhatsApp only needs to know how to send and receive messages to/from our friends; the only reason WhatsApp needs access to the plaintext *content* of our private messages is to display them to us. Therefore, we can protect WhatsApp messages with end-to-end encryption if we can isolate the UI logic for inputting and displaying messages and limit access to the plaintext messages to this component, while having the rest of WhatsApp use the encrypted messages.

The centerpiece of our solution for general and transparent data protection technology is a systems mechanism called *security overlay*, which runs as an individual app on the mobile device, and acts as a by-default transparent window that sits on top of all other applications on the system. Since the security overlay runs as a separate app, its data is automatically isolated from other apps by the system's own sandboxing policy. Furthermore, since the security overlay sits on top of all other applications, it can intercept user inputs and modify it, e.g., encrypt it, before forwarding it to the app underneath, and it can render on the overlay window the modified app output, e.g., decrypt and display plaintext for the user. Such capabilities allow the security overlay to proxy the user's sensitive data while preserving the user experience with the underlying app.

We can think of the security overlay as a proxy between a user and an app. More precisely, the security overlay is a sandbox containing the handling of private user data, starting from the point where such data is entered (by the user or otherwise) into the underlying application, to the point where the data reaches an output device (be it the network interface or the display), and guarantee that no private data is leaked to the underlying app unless it is intended by the user. The security overlay can be implemented in a wide range of scenarios. For example, it can be easily implemented as a standalone app on both Windows and Android. The security overlay can also be implemented as a device driver in OS kernel in all platforms. It can even be implemented as an application running in a secure virtual machine (VM) in a virtualized environment and become part of the trusted computing base (TCB).

UI Automation

We have glossed over an important detail in our discussion so far, that is, how can the security overlay know what and how data should be displayed, for example, what user input should be modified or encrypted, and what decrypted data should be rendered on the overlay and where it should be located. The answer to these questions lies in the UI Automation or Accessibility library that can be found on every popular platform. The UIA library on every platform we have studied provides the following general capabilities:

- It presents information about every UI component (controls, static text, etc.) of every application in the system. Such information includes the size, location, and content of a UI component.
- It presents the means to locate various UI components based on properties like

type (e.g., a button), relation with other UI components, or content (e.g., the text of a textbox, and for a button the text on it or some description of what the button does).

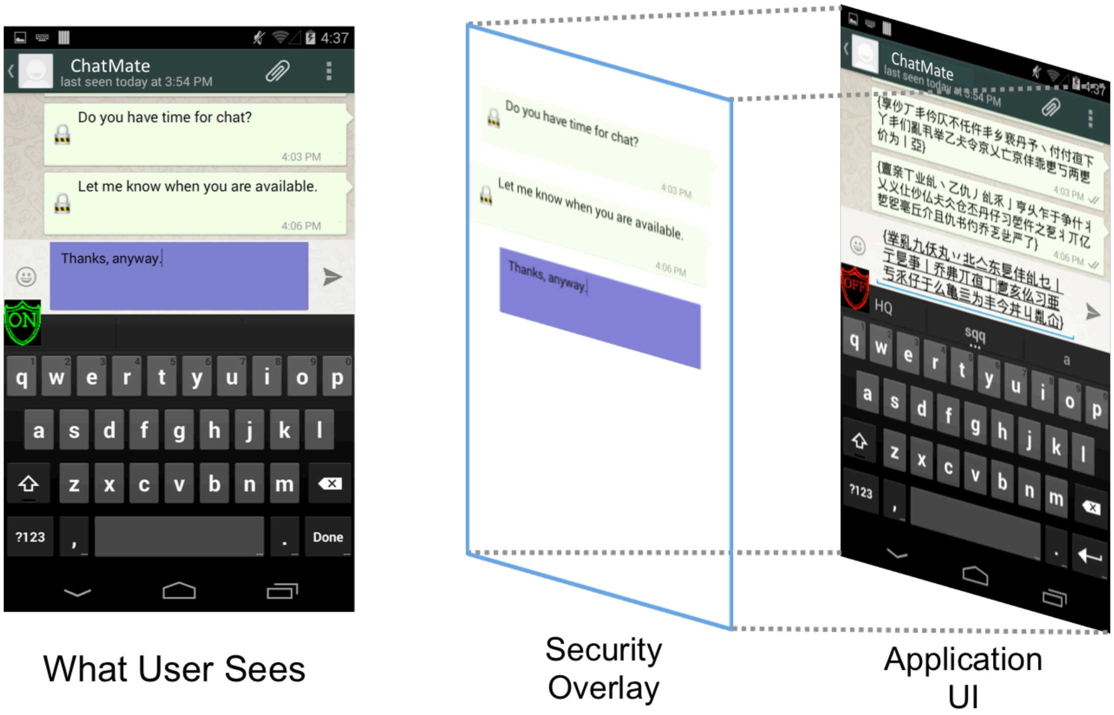
- It presents the ability to interact with any UI component (e.g., programmatically click on a button).
- It allows the monitoring of any changes to the UI (e.g., when a new UI is rendered, when focus moves from one UI component to another, or when some UI component is scrolled).

Security Overlay Mobile Apps

We implemented a security overlay app that extends the class AccessibilityService on Android and runs as a standalone app [3]. As such, the security overlay has the capabilities to monitor and probe the UI of any app; this in turn allows the security overlay to obtain the location of UI components in Gmail and WhatsApp that need to be proxied for either intercepting the user's input or decrypting ciphertext and displaying the plaintext messages. For example, by placing a functioning textbox on top of the corresponding textbox in WhatsApp that handles the message being composed, the security overlay can intercept the message the user intends to send, encrypt it and send it to WhatsApp when the user clicks the send button; similarly, by placing a textbox on top of the corresponding textbox in WhatsApp that displays an incoming (encrypted) message, the security overlay can intercept the message, decrypt it and display it in plaintext to the user. The AccessibilityService class also provides methods to control the UI components of any app; this capability allows the security overlay to programmatically interact with WhatsApp or Gmail on behalf of the user, e.g., "enter" the encrypted message in the appropriate textbox and "click" the send button. Figure 3 illustrates how security overlay transparently provides end-to-end encryption for WhatsApp.

In order to preserve application workflow, we need to enable basic functions such as search on encrypted data. Therefore, we also designed and incorporated a new searchable encryption scheme named easily deployable efficiently-searchable symmetric encryption scheme (EDESE) into security overlay that enables search over encrypted content without any server-side modification.

On iOS, it is not possible for security overlay, which is an app, to overlay a window on top of the GUI window of another app such as WhatsApp. We implemented an iOS app that is essentially a GUI shell that emulates the GUI of WhatsApp. It captures user input and sends it to an Internet server that runs our security overlay app, which runs on top the original Android version of the target app. Likewise, the GUI shell displays data sent by the security overlay app, which decrypts the output of the original Android version of the app, to the user. The server emulates Android and can run multiple versions of Android apps. From the user's point of view, the look-and-feel and workflow of the original app do not change, although the network latency has been doubled because the intermediate Internet server is involved.



ChatMate
last seen today at 3:54 PM

Do you have time for chat?
4:03 PM

Let me know when you are available.
4:06 PM

Thanks, anyway.

Figure 3 Security Overlay on WhatApp

Results and Lessons Learned

Our experiments showed that security overlay can correctly perform end-to-end encryption for the Gmail and WhatsApp apps running on Android, and the encryption incurs minimal performance overhead. We also verified that search over encrypted data works properly.

We conducted a user acceptability study where we asked participants to perform the same set of tasks on the original Gmail app and on Gmail app with security overlay. We found that no participant noticed major differences between the two experiences, and all said they would use security overlay to protect their data after we debriefed them. This is expected given that the security overlay has a transparent window over the app's GUI and performs input and output transformations, e.g., encryption and decryption, automatically.

While the effort of adding support for WhatsApp and Gmail to security overlay is straightforward (i.e., it involves analyzing the apps' intuitive and simple GUIs), it is still manual. As such, our current technology is still not scalable to support more than a handful of the most popular apps. In order to support many more apps, in the future, we will need to develop an approach for automatically extracting the information necessary for the security overlay to support a new app.

Phase II Commercialization Efforts and Results

Technologies

Gyrus is a fundamentally new, efficient and robust approach based on virtual machine monitoring techniques that uses hardware events combined with memory analysis to authorize outgoing data and network traffic only if it was initiated by a user using the computer. That is, regardless of how (old or new/unknown) malware gets on a host and what it does, Gyrus will prevent the malware from communicating with any other host, thus preventing the malware from fulfilling its malicious goals such as information theft and sending attack traffic. This represents a game-changing, “malware-oblivious” approach to counter malware attacks. Our group at Georgia Tech has been researching the underlying technologies of Gyrus for several years. Georgia Tech had filed patent application for the Gyrus framework, and Security Axioms has licensed the technologies from Georgia Tech. To the best of our knowledge, security vendors have not developed technologies similar to Gyrus.

The security overlay technology is a novel approach to data protection in that it utilizes UI Automation libraries in modern operating system to create a transparent proxy between a user and an app. It is not only an I/O proxy but also a display proxy. The key advantages of security overlay is that it does not require modification of the target app, and the users have the exactly the same look-and-feel and workflow. Our group at Georgia Tech has been researching the underlying technologies of security overlay for several years. Georgia Tech had filed patent application for security overlay, and Security Axioms has licensed the technologies from Georgia Tech. To the best of our knowledge, security vendors have not developed technologies similar to security overlay.

Industry and Market Analysis

We have also conducted studies by reading product information, whitepapers, and talking to VCs and industry executives. Studies by Damballa, Inc. (http://www.damballa.com) have shown that even the best-protected enterprise networks, with firewalls, IPS/IDS, and AV installed, still have at least 5% to 10% of their computers controlled by malware. For home computers, this figure is above 15%. This capability gap highlights that existing approaches have a fundamental limitation: malware authors have the theoretical upper hand in how they morph the malware programs and behaviors in order to limit the detection coverage of signatures and runtime behavior models. Gyrus is based on one of the few intrinsic malware properties (or “choke points”): malware needs to communicate and its data is not initiated by human. Thus, Gyrus has the potential to stop all malware that sends data/traffic.

Gyrus can co-exist with and complement other host-based and network-based security solutions. For example, Gyrus does not analyze how malware has changed the system configuration whereas an AV system can perform this task well; on the other hand, Gyrus can stop traffic by new malware even when the traffic is sent on a legitimate protocol whereas an AV or firewall typically fails. Thus, Gyrus should be marketed as an important security add-on instead of a replacement, and should follow the pricing models of existing host-based security solutions such as firewalls or AV systems.

In cloud and mobile computing environments, since end-devices can be compromised and remain to be the security weak spots, CloudCapsule can provide the needed security protection for data access. CloudCapsule complement existing cloud security technologies such as data encryption and server security monitoring.

Potentially all computers and networks can use Gyrus and CloudCapsule to improve security. On the other hand, Gyrus and CloudCapsule will be particularly effective where stopping data exfiltration and malicious traffic is of paramount concern and where access to secure networks and systems are allowed via VPN connections. Thus DoD, government, and intelligent agencies are the most likely early users of our technologies. In addition, financial institutions, and high-tech companies are likely to find our technologies very valuable.

Security overlay fills a very important gap in mobile data security. Currently, there is no general, transparent data protection for the average consumers. For example, a user would need a special messaging app or email app if he wants security protection. On the other hand, with security overlay, he can continue to use the same app and service while enjoying added security. For enterprise users, while mobile data management (MDM) solutions can secure access to data in corporate servers, there is no protection of third-party apps, which enterprise users may use for business purposes because of conveniences (e.g., chatting about business secrets on WhatsApp). With security overlay, such use of third-party apps can also be secured automatically.

Given that security overlay is a general data security solution, it is valuable to all users, including both individual and enterprise users.

Business Development

We have used aniterative process of showing prototypes to VCs and industry experts, getting feedback in particular on how our technologies can meet with market needs, and refining our technologies. For example, the mobile versions of CloudCapsule and security overlay were due to suggestions from meetings with industry experts. More specifically:

- We have created YouTube demos of both the desktop and mobile versions of CloudCapsule and have been sharing them with interested parties. We have also given several in-person demos and held several meetings with potential customers and investors. For example, we met an executive team at Wipro, Grant Wagner at NSA, Kevin Borders (formerly at NSA), John Marshall at AirWatch, Glenn McGonnigle at TechOperators, and Sudip Chakrabarti at Andreessen Horowitz.
- Wenke Lee had participated in the Value Creation Workshop and we presented at the DoD SBIR Workshops in 2013 and 2014
- We have also created YouTube demos of the security overlay mobile apps, and also show live demos to a number of VCs and serial entrepreneurs, including Dr. Paul Judge, Glenn McGonnigle, and Sudip Chakrabarti, as well as an executive team at Samsung. We also gave a presentation to Dr. Vint Cerf, considered the father of the Internet.
- We are continuing to improve and refine our technologies, and have continued dialogs with potential investors and potential partners.

We are ready to release our apps through several big social groups, including students of the on-line Masters in Computer Science program at Georgia Tech. We will initially focus on user studies and feedbacks, including the wish list of killer apps. Our strategy is to first gain a large footprint, or, user base, and then develop a business model based on metrics collected from the user base.

Lessons Learned

We learned several valuable lessons:

- We need to be very flexible and adaptive based on market needs. That is, we must be willing to refine or even reinvent our technologies. Our mobile version of CloudCapsule and the security overlay technologies are all due to feedbacks from the (potential) market.
- It is important to have a partner in charge of business development and have this partner join the team as early as possible. We were not successful in finding a good match partly because the person we had started working with changed his plan and then we started late to find a replacement.

Overall, our experiences from the commercialization efforts are very positive – we become better technologists and business executives at the same time.

Conclusion

In this project, we have studied the problem of data protection in the cloud and mobile computing environments. The goal is to provide security solutions that are attack agnostic, general, and transparent. We have successfully developed CloudCapsule for desktops and mobile devices, as well as security overlay apps for both iOS and Android. We also have started our commercialization efforts, and in particular, ready to release these products.

Bibliography

[1] Yeongjin Jang, Simon P. Chung, Bryan D. Payne, and Wenke Lee. Gyrus: A Framework for User-Input Monitoring of Text-Based Networked Applications. In Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2014.

[2] K. Borders, E. V. Weele, B. Lau, and A. Prakash. Protecting Confidential Data on Personal Computers with Storage Capsules. In Proceedings of the 18th USENIX Security Symposium, August 2009.

[3] Billy Lau, Pak Ho Chung, Chengyu Song, Yeongjin Jang, Wenke Lee, and Alexandra Boldyreva. Mimesis Aegis: A Mimicry Privacy Shield – A System's Approach to Data Privacy on Public Cloud. In Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, August 2014.