PROFESSIONAL DEVELOPMENT COURSES

M 1
Monday
12/3/2013
Full Day

Mobile Security: Securing Mobile Devices & Applications

Mr. David Lindner
Aspect Security

Mobile applications enable new threats and attacks which introduce significant risks to the enterprise, and many custom applications contain significant vulnerabilities that are unknown to the team that developed them. Considering the number of mobile applications available in the Google Play and Apple AppStore is nearing 1.5 million and vulnerabilities are skyrocketing it is imperative to perform typical application security practices. But, how is mobile different?

This one-day, hands-on course enables students to understand how easily mobile devices and applications can be successfully attacked. They will learn how to identify, avoid and remediate common vulnerabilities by learning critical security areas such as those identified in the OWASP Top Ten Mobile Risks and Controls. Using state-of-the-art testing tools, students will learn how to secure mobile applications across the enterprise. Students will be able to choose from iOS or Android hands-on labs throughout the course, while they learn how easily the bad guy can compromise applications and the data they contain.

Learning Objectives:

- Understand how mobile devices and applications can be easily attacked.
- Identify common vulnerabilities.
- Be able to use state-of-the-art mobile application security testing tools.
- Think like an attacker so that students can be preemptive.

Prerequisites.

- For Android labs: (1) Laptop with the ability to run an Ubuntu 10.04.3 Virtual Machine (Vmware); (2) CPU and memory as required by the operating system; (3) 10 GB free disk space
- For iOS labs: (1) PC running recent version of Mac OS X, with Xcode installed; (2) CPU and memory as required by the operating system

Outline:

1. **Mobile Application Risks**. Introduction to Application risks and how to emulate mobile apps and use mobile testing tools.
    a. OWASP Mobile Security Resources
    b. Current state of Mobile AppSec
    c. Top 10 Mobile Controls
    d. How and Why Attackers do it
    e. Understanding Risk
    f. Consequences
2. **Mobile Application Architectures Deeper Dive**. Different styles of computing in the mobile space, the core technologies involved, and how applications are built.
    a. Device Protections built into Android and iPhone
    b. Data Protection
    c. Encryption
    d. Client Only Architecture and Recommended Controls
    e. Client-Server Architecture and Recommended Controls
    f. Recommendation: Standard Security Controls
    g. Mobile Web Applications and Recommended Controls
    h. HTML 5 Risks
    i. JavaScript Framework Risks
    j. Same Origin Policy
3. **Mobile Authentication**. We explain how the user proves their identity to the phone, how server-side applications can authenticate the user, and how the phone can authenticate the services used.
    a. Threats: lost/stolen phone, remember me, sniffing
    b. Strong Authentication vs. User Usability
    c. Communicating credentials safely
    d. Storing credentials safely
4. **Mobile Session Management**. How to handle session management with mobile devices.
    a. What not to do.
    b. iOS and Android Recommendations
5. **Mobile Data Protection**. All of the different places that sensitive data can be stored on phones, and how it can be protected.
    a. Identifying sensitive data
    b. Threats: Lost or Stolen Devices, Sniffing
    c. Protecting data in transit
    d. Securing Communications
    e. Testing communication strength
    f. Protecting data at rest
    g. Where and how is data stored on devices
    h. Storing keys
    i. Browser Caching
    j. Mobile specific 'accidental' data storage areas

   k. Where NOT to store your data on the device
   l. HTML5 local storage

6. **Mobile Forensics**. Where application data and configuration information typically gets stored on the mobile device.
   a. Forensics tools for Android and iPhone
   b. Exploring the file system (Android / iPhone)
   c. Jailbreaking grants more access
   d. Interesting areas of the file system (Android / iPhone)
   e. Application configuration files
   f. Autocomplete records / iPhone app screen shots
   g. Dumping Android Intents
   h. Scrounging in Backups
7. **Mobile Access Control**. The code-access security models to use in mobile apps.
   a. Threat: user attacks server
   b. Example attacks
   c. Documenting your access control policy
   d. Mapping enforcement to server side controls
   e. Presentation Layer Access Control
   f. Environmental Access Control
   g. Business Logic
   h. Data Protection
   i. Hands On: Access Other Peoples Accounts, Steal Funds
8. **Other Applications**. How do we treat the threat of other applications?
   a. Risks of AppStores
   b. Malware
   c. Rooted devices and applications
   d. What can developers do?
9. **Protecting A User's Privacy**. How the phone can be used to undermine user privacy without their knowledge
   a. Using location services (GPS, cell triangulation, compass, hardware device key)
   b. Accessing contacts, photos, maps, and other personal data
   c. Accessing calls, SMS, browser, cell usage history
   d. Using camera, microphone safely
10. **Hack It and Bring It!** A hands-on challenge for students to demonstrate what they have learned.
11. **Wrap Up, Close and Thank You**

About the Instructor:

Mr. David Lindner is Aspect's Managing Consultant; Global Practice Manager, Mobile Application Security Services. David brings 13 years of IT experience including application development, network architecture design and support, IT security and consulting, and application security. David's focus has been in the mobile space including everything from mobile application penetration testing/code review, to analyzing MDM and BYOD solutions. David also specializes in performing application penetration tests utilizing commercial and freeware products as well as manual testing methods. David has written code in many different languages but specializes in Java/J2EE and Perl. David has supported many different clients including financial, government, automobile, healthcare, and retail. David holds an M.S. degree in Computer Engineering and Information Assurance from Iowa State University, recognized by the NSA as a National Center of Academic Excellence in Information Assurance Education. His Master's thesis was Creating Secure Web Applications and incorporating security throughout the Software Development Lifecycle. (SDLC). David completed his undergraduate work at Wartburg College in Waverly, IA where he received a B.A. with a triple major in Computer Science, Physics, and Mathematics.

M2
Monday
12/9/2013
Full Day

Integrating Security Engineering and Software Engineering

Dr. Antonio Maña Gomez[1] • **Michael McEviley**[2] • **Dr. Carsten Rudolph**[3] • **Mr. Jose F Ruiz**[3]
[1]University of Malaga • [2]NIST • [3]Fraunhofer SIT

Traditional practices for developing secure systems were (and still are in many cases) closer to art than to an engineering discipline. Security is still treated as an add-on and is therefore not integrated into software development practices. Experienced security artisans are still key for achieving acceptable levels of security.

Several approaches and research strands have tried to address this situation in order to introduce rigor and engineering approaches in the treatment of security aspects in information systems, mainly focusing on the final development phases or some specific aspects. Still today, one finds in the literature that the main books about security engineering describe isolated techniques and lack systematic and comprehensive treatment of security that covers the complete system lifecycle. The main drawbacks of those approaches is that they fail to provide a reasonable support for systematic engineering since the identification, characterization and specification of the protection goals and threats as well as the selection of appropriate mechanisms and countermeasures depends on the experience of the engineers. Consequently, they represent only minor improvements over the security craftsmanship era. However, they have been used for some time with uneven results.

These considerations lead us to conclude that system engineering processes must evolve in order to integrate security naturally throughout the development cycle instead of relying on it as an add-on or external component to be integrated a posteriori. In this line, recent works carried out both in the US and EU have shown that the state of the art allows us now to finally address the redefinition of security engineering into a fully-fledged proper engineering discipline and to integrate it with current software and system engineering processes. In particular, we highlight the fact that the Executive Order 13636, entitled "Improving Critical Infrastructure Cybersecurity" introduces efforts for "Building a set of current, successful approaches-a framework-for reducing risks to critical infrastructure". This course will present the most solid foundations available for building such framework, and will:

- Provide attendees with the necessary knowledge of the current situation;
- Present a global integrated vision of the security and system engineering activities;
- Present two paradigmatic initiatives for integration of security engineering into software/system engineering: The NIST initiatives for integration of security engineering and software engineering and the SecFutur approach for an integrated model-driven development process for secure systems;
- Provide guidance for the practical application of the course content by attendees; and
- Discuss the impact of the application of these initiatives in relevant emerging computing paradigms.

The central topic of the course is the integration of software and security engineering. Consequently, the overall objective of this course is to provide attendees with a clear vision, well-defined methodologies, and practical knowledge to adopt an integrated treatment of security engineering and software engineering processes in their organizations, thus improving the security of cybersystems. One important consideration from our experience is to design courses with a practical and analytic approach, ensuring that participants get the necessary perspective and can apply the knowledge in practice. We have therefore designed the course to (i) provide attendees with knowledge they can actually apply in their organizations; and (ii) avoid content-oriented approaches, and adopt instead a goal-oriented approach in which all contents are explained as tools for the main objective, which is to help attendees improve the security engineering and software engineering practices in their organizations. We define 4 main goals for the course. After taking this course, attendees should be able to:

1. Understand secure engineering principles, activities and best practices, including the role of each activity in an integrated process, and their interrelations.
2. Know the capabilities and limitations of the state of the art for each of these activities.
3. Understand the NIST and SecFutur initiatives for integration of security and software engineering and, in particular, know how to adopt these methodologies in their organizations.
4. Know the challenges of developing secure systems for emerging computing scenarios, and know how integrated methodologies can help them tackle those challenges.

In summary, by taking the course, attendees will:

- *Gain a profound knowledge of the state of the art and of methodologies and tools*. To ensure this, contents will be presented not only as a collection of facts, but also from an analytic and practical perspective.

- *Practical ready-to-use know-how*. The course contains material that will allow attendees to adopt the approaches, methodologies and tools presented in the course in their own organizations.
- *Personal consulting*. After the course, each attendee will be offered a free 30 min personal consulting session on how to better adopt the presented methodologies in their particular organizations as well as further follow-up by email. Depending on the number of attendees interested in this offer, and their preferences, we will schedule those sessions during the conference, or after it via videoconferencing.

Prerequisites. The course is designed for engineers and developers that need to deal with security aspects when designing and developing software systems. However, the content covered and the presentation strategy allow us to target a wider audience, which includes researchers, people interested in modelling, security solutions developers, etc. The course touches topics that we believe will be of interest of most attendees of ACSAC. In order to fully assimilate the topics covered in the course, attendees should have at least basic experience in system development and security. Background on security solutions and methodologies would be useful, but is not required.

Outline:

1. **Security in Systems Engineering, current approaches to SSE, and problems** (60 minutes)

   The first part of the course introduces the vision and challenges of security in systems engineering. We will explain the importance of adopting robust and integrated security engineering practices using real examples and how an integrated software and security engineering methodology can benefit developers and users by ensuring that security is adequately treated during the whole system lifecycle.

   This part of the course will be devoted to describe the state of the art and to analyze the different engineering approaches, methodologies and artifacts identifying their problems. We will cover: Threat based initiatives, Risk based initiatives, Formal methods based initiatives and Model based initiatives.

2. **Integrating Security in Systems Engineering: The big picture** (90 minutes)

   This is the first of the two central parts of the course. In this part we provide a global view of the field, and describe its activities, modeling approaches and engineering processes. Contents will deal with: Introduction and models for integration; Engineering activities: risk analysis, requirements (elicitation, specification, traceability, validation), design, secure code development & testing; Security modeling: formal modeling, security patterns, S&D patterns, UML-based approaches; Security in Architecture Frameworks; Monitoring and transparency; Compliance, Certification and Assurance.

3. **Current Cyber-security Engineering Initiatives** (180 minutes)

   This part will present two pivotal and relevant initiatives for integrating security and software engineering, describing their common and differential aspects, and the keys for their practical application. We also present other related initiatives.

   - *US: NIST Initiatives*. This part will describe the different initiatives ongoing at NIST, covering: Introduction and Objectives; Initiatives, Techniques and Standards developed at NIST; Examples of practical application; and Future Work and Challenges
   - *EU: SECFUTUR Project*. This part will describe an European initiative jointly developed by a set of selected partner organizations across Europe for setting the foundations of an integrated discipline for software and security engineering. Contents will be: Introduction and Objectives; Techniques, Artifacts and Processes; Practical application and supporting tools; and Future Work and Challenges.

4. **Relation with relevant current and future computing scenarios** (30 minutes)

   The final part of the course will revise the challenges derived from the new computing scenarios that are already taking the industry by storm such as service-oriented computing and cloud computing. In particular, we'll analyze the need for an integrated approach to security engineering and software engineering in: Embedded systems; Internet of things; Service-based systems; and Cloud Computing

About the Instructors:

Prof. Dr. Antonio Maña received his MSc and PhD degrees in Computer Engineering from the University of Malaga in 1994 and 2003, respectively. In 1995 he joined the Department of Computer Science at the University of Malaga where he is currently Professor in the Computer Science Department and leaders of the PROTEUS Research Laboratory. He is also the Research Director at Safe Society Labs. He has more than 15 years of experience working in the field of computer and software security, and on practical application of software engineering. His current research activities include integration of security and software engineering, advanced multi-layered monitoring, information and network security, security in service-based systems and cloud computing, computer-processable security certification, and software protection. He has more than 120 peer-reviewed publications. He has continuously

participated in EU funded projects since 2001. He is the Principal Investigator of the PROTEUS research laboratory in FP7 OKKAM, PASSIVE, SECFUTUR, ASSERT4SOA, CUMULUS and PARIS projects, and has previously been involved in the FP6 Serenity, iAccess and GST projects, and FP5 CASENET project. Dr. Maña is member of the editorial board and reviewer for several international journals, and participates in numerous research and education activities.

Dr. Carsten Rudolph received his PhD in Computer Science at Queensland University of Technology, Brisbane in 2001. Since then, he is working at the Fraunhofer Institute for Secure Information Technology SIT where he is now the head of the research department on Secure Engineering. His research concentrates on information security, formal methods, security requirements engineering and the integration of hardware-based security solutions. Among other activities he has worked on a security validation of the Trusted Platform Module TPM 1.2 on behalf of the German BSI and he contributes as invited expert to the standardisation of the TPM in the Trusted Computing Group TCG. Lately, he has coordinated the EU FP7 project SecFutur on security engineering for embedded systems and he is involved in various other international and German research initiatives. He also acts as a principal investigator at the Centre for Advanced Security Research Darmstadt CASED.

Mr. Jose Fran. Ruiz is a security researcher engineer at the Fraunhofer SIT. He is currently working on his PhD thesis focused on modeling artefacts for security engineering. His current research activities also include security and software engineering, privacy, information security and software evolution. He has several international peer-reviewed publications. He has worked in several European projects of the FP7 (OKKAM and SecFutur) and was previously involved in the FP6 project Serenity. He was leader of the work package 4 (Security Engineering Process) of the SecFutur project and now is working managing SIT's work in the project. He has served in the organization committee and as reviewer in different conferences and workshops and is member of several international workgroups.

M3
Monday
12/9/2013
Full Day

Introduction to Reverse Engineering Malware

Dr. Golden G. Richard III
University of New Orleans

Reverse engineering involves deep analysis of the code, structure, and functionality of software using both static and dynamic methods. This tutorial will provide attendees with a basic foundation in reverse engineering malicious software and guidance for substantially increasing the depth of these skills in the future. Reverse engineering skills are crucial in understanding modern malicious software and this deep understanding, in turn, is necessary to evaluate the impact an attack has had on a system, to recover from the attack, and to craft solutions prevent future attacks. Reverse engineering is also useful for creating interoperable software, for verifying that software patches function as promised, and for the simple joy of understanding at a deep level how software works.

This tutorial provides attendees with knowledge (and some modest experience) in reverse engineering malware, covering a range of malware types, from "historical" (e.g., DOS boot sector viruses) through modern malware. The tutorial is modeled on the instructor's experiences in teaching full-semester, highly-immersive, reverse engineering courses to undergraduate and graduate students at the University of New Orleans. The tutorial is intended to appeal to the generally curious, to researchers for whom having malware analysis skills might be useful, and to academics considering introducing reverse engineering modules into their computer security curriculum. The training includes two instructor-assisted "breakout" sessions in which teams of attendees statically analyze simple malware samples (on paper). Naturally, a one day session provides insufficient time for "mastering" even the basics of reverse engineering, but the tutorial provides a firm foundation on which to build additional skills for practice, research, and instruction. Static and dynamic analysis tools, including IDA Pro, and OllyDbg are demonstrated in the tutorial and detailed walkthroughs of malware source code reinforce the basic concepts that are introduced.

Prerequisites. Basic knowledge of assembler and systems concepts. Attendees should be either moderately comfortable with reading assembler or recall a time in which they were not completely uncomfortable doing so. The tutorial format will include time for attendees, in small groups, to tackle analysis of malware code samples (in hard copy) followed by a detailed walkthrough by the instructor. Any rust on preexisting assembler skills will be quickly sanded away. Attendees should also possess basic knowledge of systems, including compilation, linking, debuggers, concepts associated with executable file formats, etc. The course will only briefly touch upon legal issues associated with reverse engineering.

Outline:

1. **Introduction** (Brief, ~15 minutes)
    a. Course Overview
    b. Instructor Background
    c. Course Goals
    d. Overview of Legal Issues and Disclaimer
2. **Reverse Engineering Background** (1.5 hours)
    a. Why Learn / Teach Reverse Engineering?
    b. Overview of Historical and Current-generation Malware: Viruses, Worms, Trojans; Infection / Propagation strategies; Polymorphic / Metamorphic Malware
    c. Tools for Static and Dynamic Analysis: Executable File Formats; Disassemblers; Debuggers; Tools for Live Analysis: Registry Monitoring, Filesystem Monitoring, System Call Tracing
    d. Brief Refresher on Intel Assembler (w/ handouts / cheat sheets)
    e. PE Executable File Format Internals (w/ handouts)
3. **First Immersion: Malware Sample # 1** (1.5 hours)
    a. Essential Background (w/ handouts / cheat sheets)
    b. IN TEAMS: Attendees Tackle Analysis of Malware Disassembly w/ Help of Instructor
    c. Detailed Walkthrough by Instructor and Handout of Complete Solution for Further Study
4. **More Advanced Reverse Engineering: What You Need to Learn to Tackle Modern Malware** (1 hour)
    a. Encrypted / Packed Executables
    b. Anti-debugging / Anti-emulation / Anti-virtualization Techniques
    c. Code obfuscation
5. **Second Immersion: Malware Sample # 2** (1.5 hours)
    a. Essential Background (w/ handouts / cheat sheets)
    b. IN TEAMS: Attendees Tackle Analysis of Malware Disassembly w/ Help of Instructor
    c. Detailed Walkthrough by Instructor and Handout of Complete Solution for Further Study
6. **Summary / Wrap up / How to Develop Deeper Skills** (Brief, ~15 minutes)

About the Instructor:

PROFESSIONAL DEVELOPMENT COURSES

Dr. Golden G. Richard III is Professor of Computer Science, University Research Professor, and Director of the Greater New Orleans Center for Information Assurance (GNOCIA) at the University of New Orleans. At UNO, he teaches courses in digital forensics, reverse engineering, offensive computing, operating systems internals, and malware analysis, which are also his current areas of active research. Golden is a member of the United States Secret Service Electronic Crime Taskforce, holds a position on the Editorial Board of the Journal of Digital Investigation and the International Journal of Digital Crime and Forensics (IJDCF), is a member of the American Academy of Forensics Sciences (AAFS), a member of the ACM, IEEE, and USENIX, and serves as the academic liaison for USENIX to UNO. He is also a founding member and chairman of the non-profit that runs the Digital Forensics Research Workshop (DFRWS), the premiere venue for publishing digital forensics research. He earned a B.S. in Computer Science from the University of New Orleans and M.S. and Ph.D. degrees in Computer Science from The Ohio State University.

Dr. Richard has given tutorials at ACSAC, IPCCC, Mobicom, PDCS, USENIX ATC and USENIX Security on a variety of topics, including digital forensics, reverse engineering, and (in the more distant past) mobile computing concepts and service discovery protocols. He published a paper in CSET 2009 detailing his approach to teaching reverse engineering in academia, which, subject to depth and time limitations, underlies his approach in the proposed training.

T4
Tuesday
12/10/2013
Full Day

Analysing Android Malware at Runtime

Dr Giovanni Russello
University of Auckland

Android-based smartphones are the most sold in the world dominating the market share with a solid 72.4% [1]. A key-aspect in Android's success is the support for third-party applications (or simply apps) creating a very dynamic software landscape accessible through the Google Play marketplace as well as third-party markets.

The rate of Android success is only matched by the increase in malicious activity targeting Android. Between 2011 and 2012 the malware samples targeting Android has gone up of 1000% [2]. At the end of 2012, Android has crashed another record becoming the top target for malicious code overtaking Microsoft's Windows operating system [3].

Android is not only dominating the mobile device market (smartphones and tablets), but is also becoming predominant in mission critical support and infotainment car systems. The implication of its security issues can be very important in these sectors as well. For instance, through Android malware could find its way to interact with the Can Bus system of a car.

In this course, we will study the security model in Android and how malware is able to bypass some of its security features. To better understand the security exploits, the first part of the course will be dedicated to the Android security framework and how apps interact with it. The second part of the course will focus on the analysis of real malware samples. To demonstrate the malware capabilities, we will use a real Android device where the malware samples will be installed and executed. By means of a tracing tool developed in our lab, we can monitor at runtime the malware execution and display its action to the audience. Finally, we will cover recent research effort in securing the Android OS.

Prerequisites. An understanding of Operating Systems (Linux in particular) and Access control models (MAC and DAC).

Outline:

1. **Introduction** (1 hour)

   An initial overview of the course content followed by an overview of the basic principle of system security to bring all the students at the same level of knowledge on access control and policy-based systems.

2. **Overview of the Android Security Framework And Inter Component Communication (ICC)** (1 hour)

   We will dive in the details of the security framework of Android and some of its not-so-well documented exceptions/refinements. To better understand some of the malware action is also important to cover the ICC mechanism offered by Android to apps for exchanging information and communicate with the system services (e.g., SMS sending service).

3. **State of the art** (1 hour)

   We will discuss the state of the art in research, covering the most recent research efforts in security for the Android OS. We will also discuss why current commercial solutions, such as Anti-Virus Software are not capable of contrasting this huge wave of attacks.

4. **Malware Classification** (1 hour)

   There are several malware families for Android. We will discuss each of these families providing details of their malicious actions, and what damage/loss they cause.

5. **Malware Runtime Dissection** (1½ hour)

   In this part of the course, we will use a real device (connected to the projector) where several malware samples will be deployed (at least one for each malware family). By means of an analysis tool developed in our department, we will trace the actions performed by the malware at runtime showing the details of each attacks.

6. **Malware Runtime Dissection** (½ hour)

   We will conclude with some final remarks, detailing some techniques that can be used to protect apps developed for this platform with emphasis on mission critical and cyber-physical systems.

About the Instructor:

Dr. Giovanni Russello is a lecturer at the University of Auckland. He has worked on policy-based systems, access control mechanisms, and cloud security for more than 10 years. In the last two years, he has focused his research efforts in enhancing security for the Android OS. This professional development course is based on Giovanni's experience in the field. Giovanni has already provided a longer version of this course as a postgraduate course at his department receiving excellent feedback from his students. Giovanni is the founder and CEO of Active Mobile Security, a stealth startup focusing on mobile security.

PROFESSIONAL DEVELOPMENT COURSES

T5
Tuesday
12/10/2013
Full Day

Finding Data Leaks in Applications, Network Protocols, and Systems with Open Source Computer Forensics Tools

Dr. Simson Garfinkel
forensicswiki.org

Many kinds of data leaks and security flaws are easy to find if you just look. Hard-coded usernames and passwords, weak or missing cryptography, and logfiles containing inappropriately sensitive information are easy to spot—provided that you know what sensitive data looks like, and provided that you're using the right tools. Although many privacy and security auditors restrict themselves to reading privacy policies, written specifications and architectural diagrams, experienced investigators know that there is no substitute for looking at the data as well. This is especially true of cyber-physical systems, which were historically developed by programmers that have little training in information security. Frequently these developers don't realize that security snafus hiding in their own code.

This course teaches how to use the open source tools bulk_extractor and tcpflow to analyze application files, databases, network packet traces, memory dumps, and entire operating systems for data leaks and other kinds of related security problems. It teaches the student how to recognize sensitive data when encoded in a variety of different formats, and how to extend the open source tools when presented with data that is in a proprietary form. (Such proprietary formats are common with process control systems.) It presents famous cases of how sensitive data was left behind by application programs, operating systems, programmers and users in PDF files, databases, network connections, and system memory. Finally, it presents programming patterns for eliminating leakage of sensitive information.

Prerequisites. Basic knowledge of scripting languages (e.g. python) and cryptography (hash functions, symmetric algorithms such as AES, asymmetric algorithms such as RSA, and PKI).

Outline:

1. **Introduction**
   a. Auditing for Data Leakage: (1) What is Data Leakage? (2) What is Auditing? (3) Advantages: You can find stuff that the vendor / designer / programmer doesn't tell you about/doesn't know about. (4) Limitations: You can't find it all. (5) Kinds of auditing: black box / grey box /white box
   b. What are we looking for? (1) PII - Personally Identifiable Information (email addresses, names, CCNs, etc) (2) Plaintext passwords. (3) Hardcoded passwords (4) Hardcoded URLs, IP addresses (5) Examples
   c. What Data Look like: (1) ASCII (2) Unicode (3) Hex Dumps (4) Strings (5) Numbers (6) Magic Numbers (7) Encodings (base16, Base64, Base85, compression)
   d. Analyzing Application Programs at Rest: (1) Strings; (2) Disassemblers for x86, ARM and Java; (3) Why you want to avoid disassembling
2. **Bulk Extractor**
   a. Introduction - Using bulk_extractor: (1) What it is; (2) How it works
   b. Using bulk_extractor: (1) To analyze programs and program installations; (2) To analyzing running computer systems; (3) To analyze memory
3. **Networks**
   a. Introduction: (1) Brief introduction to IP networks, TCP protocols, and Encryption Protocols; (2) Live vs. captured analysis; (3) Packet file formats; (4) Wireless vs. Wired networks; (5) Making network captures with tcpdump; (6) Wireshark to analyze individual packets; (7) Tcpflow to analyze TCP streams
   b. Using tcpflow: (1) Breaking a packet capture into
   c. Dealing with encryption: (1) Decrypting SSL with decrypting proxies; (2) Decrypting SSL with server keys
4. **Extending these tools**
   a. Post-processing with python modules
   b. Working with Unicode and large files with Python
   c. Writing extensions for bulk_extractor and tcpflow in C++

About the Instructor:

Dr. Simson L. Garfinkel is an Associate Professor at the Naval Postgraduate School. Based in Arlington VA, Garfinkel's research interests include computer forensics, the emerging field of usability and security, personal information management, privacy,

information policy and terrorism. He holds six US patents for his computer-related research and has published dozens of journal and conference papers in security and computer forensics.

Garfinkel is the author or co-author of fourteen books on computing. He is perhaps best known for his book Database Nation: The Death of Privacy in the 21st Century. Garfinkel's most successful book, Practical UNIX and Internet Security (co-authored with Gene Spafford), has sold more than 250,000 copies and been translated into more than a dozen languages since the first edition was published in 1991. Garfinkel received three Bachelor of Science degrees from MIT in 1987, a Master's of Science in Journalism from Columbia University in 1988, and a Ph.D. in Computer Science from MIT in 2005.

Garfinkel is the primary developer and maintainer of bulk_extractor and tcpflow, the two primary tools that will be used in this course.

T6
Tuesday Morning
12/10/2013
Half Day

Authentication & Authorization Standards for the Cloud

Dr. Hassan Takabi
University of North Texas

This course aims to introduce different technologies available for single sign on and federated identity in cloud environments. We also cover existing and emerging authorization technologies in the cloud. Specifically, we will look at OAuth 2.0 as a lightweight approach for authorization for RESTful services and application. We review through some use cases what benefits it provides and how it can be integrated with other technologies like SAML 2.0 to provide integration, federation and interoperability in cloud computing environments. We will also introduce the System for Cross-domain Identity Management (SCIM) specification which is an ongoing effort designed to make managing user identity in cloud based applications and services easier. Then, we cover the IEEE Standard for Intercloud Interoperability and Federation (SIIF) which is an ongoing effort for cloud-to-cloud interoperability and federation. Finally we will look at efforts undertaken by government agencies regarding authentication and authorization in the cloud.

Prerequisites. No specific prerequisite is required. Being familiar with general security concepts, authentication, and authorization is enough.

Outline:

1. **Single Sign On (SSO) Technologies for cloud computing** (20 min)

   An introduction to various SSO technologies that are being used or are emerging as de facto standard will be provided.

2. **The Security Assertion Markup Language (SAML) 2.0** (20 min)

   SAML is an XML-based framework for communicating user authentication, entitlement, and attribute information. A high-level overview of SAML will be given followed by a technical introduction to SAML concepts and capabilities.

3. **The OAuth 2.0 Authorization Framework & Use Cases** (30 min)

   The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf. An overview of the OAuth 2.0 is given with details of various flows and a comparison between flows. We will also discuss some OAuth use cases to show its applicability in real world and demonstrate how enterprises can use OAuth for authorization and how to choose the best flow based on scenarios.

4. **SAML 2.0 Bearer Assertion Profiles for OAuth 2.0 and its integration with OAuth 2.0** (30 min)

   We discuss the use of a SAML 2.0 Bearer Assertion as a means for requesting an OAuth 2.0 access token as well as for use as a means of client authentication. We will also discuss how to use SAML and OAuth 2.0 together to achieve best integration and management simplicity in identity and policy domains.

5. **System for Cross-domain Identity Management (SCIM)** (30 min)

   The System for Cross-domain Identity Management (SCIM) specification is designed to make managing user identity in cloud based applications and services easier. The specification suite seeks to build upon experience with existing schemas and deployments, placing specific emphasis on simplicity of development and integration, while applying existing authentication, authorization, and privacy models. An overview of the specification will be given along with benefits it provides and some use cases. We will also discuss in detail how to bind the System for Cross-domain Identity Management (SCIM) schema to the Security Assertion Markup Language (SAML).

6. **IEEE Standard for Intercloud Interoperability and Federation (SIIF)** (20 min)

   This standard defines topology, functions, and governance for cloud-to-cloud interoperability and federation. Topological elements include clouds, roots, exchanges (which mediate governance between clouds), and gateways (which mediate data exchange between clouds). Functional elements include name spaces, presence, messaging, resource ontologies (including standardized units of measurement), and trust infrastructure. Governance elements include registration, geo-independence, trust anchor, and potentially compliance and audit.

7. **Government efforts** (30 min)

We briefly review the cloud authentication & authorization approaches recommended by various agencies such as the Federal Risk and Authorization Management Program (FedRAMP), DoD's Cloud Computing Strategy and NIST's Guidelines on Security and Privacy in Public Cloud Computing.

About the Instructor:

Dr. Hassan Takabi is an Assistant Professor in the Department of Computer Science and Engineering and member of the Center for Information and Computer Security (CICS) at the University of North Texas. He received his PhD from the University of Pittsburgh and his research interests include access control models, trust management, privacy enhancing technologies, usable security and privacy, and security, privacy, and trust issues in cloud computing environments and online social networks. He is member of IEEE and the ACM.

PROFESSIONAL DEVELOPMENT COURSES

T7
Tuesday Afternoon
12/10/2013
Half Day

Cyber-Physical Systems Security

Dr. Alvaro A. Cardenas
University of Texas, Dallas

This class covers the security of cyber-physical critical infrastructure systems (such as transportation networks, oil pipelines, and the power grid) from a multidisciplinary point of view, from computer science security research for critical infrastructure, to public-policy, the Executive Order 13636, risk-assessment, business drivers, and control-theory methods to reduce the cyber risk to critical infrastructures.

Prerequisites. None.

Outline:

1. **Introduction** (1:00 hour)
   a. Cyberconflict: (0:30 mins)

      The first part will consist on the definition of the problem, and will look at a multidisciplinary view of potential attacks to cyber-physicial critical infrastructures. The first 30 minutes will define what cyberconflict is (including Cyber-Activism, Espionage, Terrorism, and War), and the different interpretations.

   b. Business Case (0:30 mins)

      The second part of the introduction session will also be 30 minutes, and will discuss some of the difficulties for securing our critical infrastructures, including the business cases and the recent Executive Order 13636 and the Cyber-Framework. (Note that the 4th Cyber-Framework Workshop will be hosted at the institution of the course instructor.)

2. **Security of CPS systems** (1:30 hours)

   The second part of the course will discuss the security of different cyber-physical systems including transportation networks, air-traffic control, building automation and HVAC, smart grids, and SCADA security.

3. **Hands on lab.** (0:30 mins)

   The final part of the course will be 30 mins and will give hands-on experience to students on a practical tool. It can be a physical simulation tool such as GridLAb, or it can be a discussion of SCADA Systems Connected to the Internet and some small experiments with SHODAN

About the Instructor:

Dr. Alvaro A. Cárdenas is an Assistant Professor at the University of Texas at Dallas. He holds M.S. and Ph.D. degrees from the University of Maryland, College Park under the advise of John Baras, and a B.S. from "Universidad de los Andes." Prior to joining UT Dallas he was a research staff with Fujitsu Laboratories of America in Sunnyvale, CA, where he did research in Trusted Computing, Big Data, and Smart Grids; and prior to Fujitsu, he was a postdoctoral scholar at the University of California at Berkeley under the advise of Shankar Sastry and Doug Tygar, doing research in cyber-physical systems security and applications of Machine Learning to Intrusion Detection.

He has also been an invited visiting professor at the University of Cagliari in Italy where he taught a class on machine learning and game theory for security, an Intern working on discriminatory Bayesian networks at INRIA-LORIA in France, and a SCADA intern, working on ladder logic to replace old relay boxes in Occidental Petroleum Corporation in Caño Limón, Cobeñas, Colombia. He has received numerous awards for his research including a best paper award from the U.S. Army Research Office, a best presentation award from the IEEE, a graduate school fellowship from the University of Maryland, and a Distinguished Assistantship from the Institute of Systems Research.