SECURITY EXPLORATIONS

CORE COMPETENCES
AND
APPROACH TO SECURITY

Adam Gowdiak, Security Explorations

INTRODUCTION

PRESENTATION GOAL

- Present Security Explorations
  - Our competences in the area of software security
    - experience
    - core skills
    - achievements
    - services
  - Our approach to security
    - methodology and processes

SECURITY EXPLORATIONS

ABOUT THE COMPANY

- Security and vulnerability research company from Poznań, Poland
- Various services in the area of security and vulnerability research
  - Breaking security of things and analyzing software for security defects
  - New attack and vulnerability exploitation techniques
- Quality, unbiased, vendor-free and independent security and vulnerability research

SECURITY EXPLORATIONS

PEOPLE

- Adam Gowdiak
  - Company founder and CEO
  - M. Sc. in Computer Science from Poznan University of Technology (1994-1999)
  - Poznan Supercomputing and Networking Center (1996-2005)
    - Government research facility
    - Security engineer and systems analyst
  - LSD Research Group (1996-2004)
    - Non-profit organization
    - Co-founder, principal researcher
  - Sun Microsystems Laboratories (2005-2008)
    - Commercial research laboratory
    - Senior Staff Engineer
    - Hired by Whitfield Diffie (Sun CSO, the inventor of public key cryptography)

SECURITY EXPLORATIONS

OUR EXPERIENCE

- 20+ years of contribution to the security research field
  - Security vulnerabilities and exploitation techniques
  - Research papers and conference presentations
- Bug hunting and exploit code development for various operating systems and architectures
  - Windows, Linux, Java, AIX, IRIX, Solaris, Nokia S40 OS
  - X86, MIPS, ARM, PowerPC / POWER, Sparc, SH4
- Reverse engineering
  - Security analysis of binary programs
  - Static and runtime program analysis
  - Custom tools
- Penetration testing

SECURITY EXPLORATIONS

OUR PAST ACHIEVEMENTS

- Java VM security
  - Research paper from 2002 (Sun, Netscape, Microsoft)
    - "Java Security Vulnerabilities and their exploitation techniques"

      „a 50-page paper that exposed implementation vulnerabilities in Java – far better than anything produced by the l0pht", Chris Wysopal, L0pht member
  - 24 bugs in J2SE from 2005/2006
    - Java RMI weakness, Java Reflection bug class
- Mobile Java (J2ME) security
  - Two bytecode verifier issues from 2004
    - 250 million handsets affected
- Other Java based software
  - Apple Quicktime (10+ bugs)
  - Local and remote Solaris OS issues

SECURITY EXPLORATIONS

OUR PAST ACHIEVEMENTS (2)

- Windows MSRPC DCOM
  - Critical security vulnerability in all available Microsoft Windows operating systems (2000 / XP / 2003)
  - Described in MS03-026 security bulletin
  - Remote attackers could get unauthenticated access to remote Windows systems with administrative privileges
  - Bug exploited by the Blaster worm
    - 20+ millions of systems infected
  - Focused security researchers on MS Windows RPC area
    - years later bugs were still found in MSRPC

SECURITY EXPLORATIONS

CORE SKILLS

- Strong analytical skills
  - Ability to discover issues missed by big software vendors and their security teams in software security assurance / development lifecycle
    - Vulnerabilities in products already deployed to the market
    - Broken patches
    - Hardware issues
      - STi7100 / STi7111 processors
  - Ability to discover information about a target from little scratches (puzzles as in our company logo)
    - SLIM Core instruction set reverse engineering from the format of a single SLIM Core instruction (JMP)

SECURITY EXPLORATIONS

CORE SKILLS (2)

- Systemic and in-depth security analysis
  - SE-2011-01 SAT TV research
    - CSS bug in web application code
      - 50 bytes of arbitrary HTML code
    - JavaScript code execution
    - Java code loading and execution
    - Java Virtual Machine sandbox escape
    - Native memory access
    - Native code execution
    - Kernel level code execution
    - STi7111 SLIM Core code execution
    - CW extraction

SECURITY EXPLORATIONS

CORE SKILLS (3)

- Creativity
  - Custom tools for static / dynamic code analysis
    - SH4 emulator with Crypto Core I/O proxy for set-top-box boot loader decryption
    - SLIM Core tracer
  - Novel exploitation techniques
    - Minor Java bugs chaining for complete sandbox compromise
    - Type confusion for memory access or privilege elevation
    - JVM internals for native code execution
    - Java sandbox escape for Oracle DB privilege elevation
    - …

SECURITY EXPLORATIONS

CORE SKILLS (4)

- 20+ years experience in breaking security of closed software
  - Strong reverse engineering skills
- Ability to break security of targets not known prior to the engagement
  - Hacking from scratch

SECURITY EXPLORATIONS

CUTTING EDGE SECURITY RESEARCH

- We were the first to break security of:
  - Java for mobile phones (J2ME) with MIDP 2.0 security features aimed at protecting users and devices from malicious software
  - Nokia Series 40 Platform devices
  - digital satellite TV set-top-boxes running Java MHP middleware from Advanced Digital Broadcast
  - secure cryptographic processors from STMicroelectronics used to secure HDTV content broadcasted by various SAT TV operators around the world (STi710x and STi7111 DVB chipsets)
  - Java based cloud hosting environments coming from Oracle and Google (Oracle Java Cloud Service and Google App Engine for Java),

SECURITY EXPLORATIONS

CUTTING EDGE SECURITY RESEARCH (CONT.)

- We were also the first to:
  - discover and implement an attack against a mobile 3G phone allowing for a remote deployment and execution of a malicious Java application (i.e. a backdoor, malware or virus),
  - demonstrate novel techniques for both a setup and exploitation of type confusion vulnerabilities in Java environments,
  - demonstrate novel techniques for a security compromise of Oracle Database with the use of Java security vulnerabilities.

SECURITY EXPLORATIONS

BUGS STATISTICS

Vendor	Target	#Issues
ADB	Set-top-box SW	22
APPLE	Apple Quicktime for Java	2
GOOGLE	Google App Engine	41
IBM	Java SE	26
NOKIA	Series 40 mobile phones	14
ORACLE	Java SE	44
ORACLE	Oracle Java Cloud Service	30
ORACLE	Oracle Database JVM	22
ST	STi7100 / STI7111 DVB chipsets	4
ORACLE	Java Card	31

SECURITY EXPLORATIONS

WHAT OTHERS HAVE BEEN SAYING

- The Register
  - „We reported Gowdiak's claims earlier this month, with some incredulity. It seemed unlikely that one researcher could uncover such a litany of security flaws in such a popular platform...- but it seems our cynicism was misplaced"
- Mark Durrant of Nokia's corporate communications
  - „This requires deep technical skills. This isn't something someone in a garage is going to be able to sort out in an afternoon. He's [Gowdiak's] clearly a smart guy"
- Undisclosed Israeli company
  - „Having personally seen and evaluated your publications as part of Security Explorations ... I trust you can deliver the high quality results we are looking for"
- Undisclosed US Gov / Mil contractor
  - „I remain impressed by what the Polish brain can produce"
- Google Security Team
  - „The VRP panel was really impressed by your research and thoroughness"

SECURITY EXPLORATIONS

SERVICES

- Services offering focused on our best skills and experience
  - Security evaluation of software
  - Custom security research projects
    - Binary or source code
- For software / hardware vendors and 3rd party companies
  - Is software / hardware / technology we develop secure ?
  - Is software / hardware / technology our company use secure ?
- For GOV / MIL sector
  - Offensive capabilities development
  - Intelligence acquisition
- For financial, telecommunicationon and transportation industries
  - Java Card evaluation
- Competitive and flexible pricing

SECURITY OF SOFTWARE

MYTHS

- Hackers go after Oracle, Microsoft, Apple and others
- Proprietary systems are secure
  - Secret, difficult to reverse engineer and hack (security through obscurity)
- EULA and shrink wrap licenses are sufficient to stop any hacking attempts
- Competences in HW security space are reflected in SW security space

SECURITY OF SOFTWARE

HOW SECURITY RESEARCHERS CHOOSE THEIR TARGETS ?

- Challenge
- Novelty of the research
- Impact of potential discovery
  - Major market player
  - Large number of users

SECURITY OF SOFTWARE

COMMON APPROACH

- No need to invest in security as long as no problems arise
  - No problems with security so far - no need to waste money on security
- Security as an after-thought thing
  - we'll deal with security later
- Shallow security product reviews
  - Design and architecture
  - Functionality testing instead of security testing

SECURITY OF SOFTWARE

COMMON IMPLICATIONS

- Security bugs are expensive
  - NIST estimates that the costs of fixing a bug after product release are 30x higher than if it was fixed during coding / testing phases
  - Even higher costs in the mobile / hardware world
    - The cost of patch deployment into millions of devices
    - Not so clear who should pay these costs
  - Development resource put into bug fixing
  - Security bugs need to be fixed

SECURITY OF SOFTWARE

COMMON IMPLICATIONS (2)

- Bad PR / media headlines
  - Security is a hot topic these days
  - Not all medias pay attention to the details
    - Sensation in the first place
- Potential lost of credibility and clients’ trust
  - Some big organizations (gov, mil, network operators) do pay attention to security

SECURITY OF SOFTWARE

REAL LIFE IMPLICATIONS (ORACLE)

- Apple, Google, Microsoft and Mozilla blocked Java in their web browsers
- US Department of Homeland Security warned users about Java security risks
- Certain financial institutions decided to move away from client side Java (Applets)
- US Federal Trade Commission's investigation against Oracle over deceptive Java security updates

SECURITY OF SOFTWARE

REAL LIFE IMPLICATIONS (STMICROELECTRONICS)

- ADB / Platform N choosing BCM chipsets for the set-top-boxes of a new, merger company (NC+) following our ST vulnerabilities disclosure
- 1400 layoffs and shutting down of the whole ST set-top-box business in 2016

SE APPROACH TO SECURITY

METHODOLOGY

- Approach a given target from an attacker's point of view
  - Focus on untrusted user input
    - Attackers can influence system's behavior via malicious, specially crafted input data
- The difference
  - Attacker needs to find one bug
  - Security evaluator needs to find all of them

SE APPROACH TO SECURITY

THE PROCESS

- Learning as much as possible about the target of a security evaluation
  - Technical documentation
  - Source / binary code analysis
  - Playing with the target
- Create threat model and identify the attack surface
- Select potential weak points

SE APPROACH TO SECURITY

THE PROCESS (CONT.)

- Develop and verify attack scenarios
  - Detailed source code review
  - Proof of concept codes
  - Custom tools
- Refinement phase
  - Change of assumptions / requirements
- Final report

SE APPROACH TO SECURITY

THREAT MODEL AND ATTACK SURFACE

- Identification of components directly exposed to attackers
  - Components that receive or process data from untrusted sources
    - i.e. WWW server, SIP server, SMS parser, JPEG parser, …
- Identification of components indirectly exposed to attackers
  - Components that receive untrusted data from other components
    - Web browser, image parsing library, …
- Identification of privileged components

SE APPROACH TO SECURITY

THREAT MODEL AND ATTACK SURFACE (2)

- Identification of authentication and authorization mechanisms implemented
  - How access to sensitive resources is implemented
  - Which components implement it
- Enumeration of components interaction
  - Information flow in the system
  - Mutual trust
  - Communication mechanisms used

SE APPROACH TO SECURITY

THREAT MODEL AND ATTACK SURFACE (3)

- Identification of requirements to break security of a given component
  - Start with minimal security assumption
    - Components directly exposed to attackers
    - Components without authentication / authorization
  - Follow information flow in the system
    - Components processing attacker’s data
  - Refine security assumption
    - Assumption of a component compromise
  - Repeat the process

SE APPROACH TO SECURITY

ATTACK SCENARIOS

- Attack scenarios are developed with respect to the identified requirements for breaking security of a given component
  - Feasibility of attacks verified with the use of source code review
    - Can the attack be launched ?
    - What input data needs to be used ?
  - Proof of concept codes for ideas / attacks verification
  - Custom tools for speed and automation

SE APPROACH TO SECURITY

SOURCE CODE REVIEW

- Conducted for components identified by a threat model
- Hunt for design and implementation bugs
  - Known classes of vulnerabilities
    - Memory corruption vulnerabilities
    - Injection vulnerabilities
    - Path traversal
    - Race condition
    - …
  - Manual, line by line code analysis
  - Focus on untrusted user input, its processing and propagation into other components
- Discovery of new attacks

FINAL

Q & A

THANK YOU

contact@security-explorations.com