# AFFIDAVIT OF MARIO D. SANTANA

STATE OF VIRGINIA )
) SS.
COUNTY OF FAIRFAX )

MARIO D. SANTANA, CISSP, CISA, being duly sworn, hereby swears under penalty of perjury that:

1. I am employed by Terremark, Inc. ("Terremark") as a Director of Secure Information Services, and previously by SteelCloud, Inc. as a Senior Security Consultant.
2. I graduated from Colorado Technical University in 2006 and received additional training in computer programming, systems, networks and related areas from the SANS Institute.
3. I am a Certified Information Systems Security Professional ("CISSP") and a Certified Information Systems Auditor ("CISA").
4. I have worked extensively in the field of computer programming and system development. I have also guest lectured on these subjects at Florida International University. I have written several whitepapers on software and software development. My Curriculum Vitae is attached as Appendix A.
5. Terremark was engaged to assist CMI, Inc. ("CMI") by providing a programming and systems expert to review certain source code (the "Code") developed and owned by CMI used in a particular model of CMI's Intoxilyzer device and to render an independent opinion as to whether a qualified programmer could understand from reviewing the Code the Code's logic and how the Code operates. In addition, Terremark was asked to evaluate the mechanisms by which the Code is controlled as it is passed from the development of the Code through installation into the Intoxilyzer device.
6. I was assigned to perform the required analysis of the Code and how it was controlled.
7. On Thursday, October 16, 2008, I spent a day at CMI's offices in Owensboro, KY to carry out this assignment.
8. I was presented with a printed book containing 1,116 pages of source code (the "Code Book") and an electronic document of identical content in PDF (Portable Document File)

format (the "Code PDF.") I determined that the Code Book and the Code PDF were identical by comparing the number of pages and by visually inspecting the contents in both the book and the PDF for a series of randomly-chosen pages, specifically 1-5, 63, 128, 307, 357, 630, 643, 735, 908, and 1,114-1,116.

9. There are two distinct parts to the Code. The first part is written in Z80 assembly language and pertains to the function of the general-purpose Z80 computer chip, which in general terms controls the user interface and related functions of the Intoxilyzer device. The second part is written in the "C" programming language and pertains to the 8051 chip, which in general terms controls the sensors in the Intoxilyzer device. These two parts are designed to work together as a single system.

10. Hard copy or printed pages, such as the Code Book I reviewed, can be tracked and managed more easily for security purposes than electronic copies. Of popular electronic formats, PDF is one of the easiest to control, and a single electronic document file, such as the PDF I reviewed, is easier to control than multiple electronic document files. Nevertheless, extreme care must be taken with any electronic format in order to protect its contents from being inadvertently copied and distributed. Once an electronic file is written to disk, it's difficult to erase completely without specialized tools and techniques. Unlimited identical copies may be quickly and conveniently made of an electronic document without any loss of fidelity, and without any audit trail of such copies. Where photocopying of more than 1,100 pages of paper would require significant time, and would be greater than the size of 2 reams of paper, PDF files are simply computer data and can be copied very quickly, stored on any of a number of storage devices (hard drives, flash memory devices, CDs, DVDs, etc.) Because of these and other reasons, it is extremely difficult - and often impossible - to account for every reasonably possible copy made of an electronic document while it was not under the strictest access controls.

11. The Code was inspected manually because, while there are automated tools and techniques to gather statistical information about computer source code, there is no automated method of analyzing source code to verify its function. Manual inspection is, therefore, the only reliable way of reviewing source code to understand exactly what it does and how it does it.

12. Inspection of computer source code requires a person with expertise in the particular computer languages used (here, Z80 assembler and "C".)

13. While I have the requisite expertise in Z80 assembler and "C", I did not have any prior exposure in regard to CMI's particular Source Code or explanation of the Source Code provided by the client.

14. I began by sampling random sections of the code, and then selected four functional components to analyze in more depth. The functional components I analyzed are those that implemented the calibration and diagnostic functions, the subsystem interface between the Z80 and the 8051 chips, and the main logic loops. In all cases, the source code for these components was easy to find, read and understand.

15. The computer source code I analyzed is easily readable and understood by a programmer experienced in both assembly-language programming for the Z80 computer chip and the C programming language. The following characteristics of the code lead me to this conclusion:

    15.1. The Courier font in which the book and the PDF are typeset is clearly legible.

    15.2. Comments, which are not computer instructions but rather meant for humans, are used liberally in the code to document its function.

    15.3. Program logic is divided into blocks, which are clearly delineated with comments.

    15.4. Program logic is hierarchical, with blocks of program logic nested within each other. Nesting is clearly and consistently indicated throughout the code in the usual manner for showing such nesting, which is four spaces of indentation for each level of nesting. This helps make the code easy to read for humans.

    15.5. High-level program architecture is very data-driven, meaning that the organization of data in the programs is an important part of how the program is designed. These data structures are defined and heavily commented early in the source code files, providing valuable insight into the program's design.

    15.6. Labels are declared and commented early, and used throughout the code in place of numerical constants. For example, memory addresses are referenced with labels, rather than with raw numerical memory addresses. This dramatically improves readability.

15.7. The overall software architecture is based on a straightforward monolithic design. There are none of the confusing complexities of distributed, multi-threaded or re-entrant code, because these complex features are not used by the software. This makes for source code that flows easily from one task to the next in simple, logical steps.

15.8. Where program logic begins to display any complexity, it is divided into a series of simple steps and heavily commented.

15.9. Wherever there is any subtlety of logic, these are identified and explained in comments. This applies even for standard language idioms such as control loops, which are prone to so-called "off-by-one" errors, and which are not usually commented in other source code.

15.10. Being that the main constraint of the embedded platform on which this code runs is storage space for program instructions and data, various optimization techniques are used to reduce the size of the compiled program that must be copied to and fit in the limited memory of the hardware platform. These optimizations include compiler pragmas, and the use of relative rather than absolute jump instructions. However, none of these optimizations complicate the reading or understanding of the code by a human.

16. In order to develop an opinion about whether the Code I reviewed is actually the code that shipped on a particular device, I toured the parts of the plant where the Intoxilyzer devices are developed, tested and built, and reviewed the processes used in each step. The steps to create a complete system from hardware and source code are simple and robust. Multiple checks and balances are built into the process to help eliminate errors in the manufacturing process. The steps were explained and, where possible, demonstrated for me:

16.1. Software engineering labels a completed code revision with a unique name.

16.2. Each code revision incorporates a mathematically unique checksum of its constituent code. This checksum is recalculated and verified at various stages of the development and manufacturing process to ensure the integrity of the software as it moves from step to step in the process.

16.3. This code revision is used by a quality control engineer to build a complete device, and that device is thoroughly tested for proper function, using a detailed testing plan that I was informed can take days to execute.

16.4. If testing is successful, the code is released to the manufacturing process, where it is used in the production of new units, and in the production of chipsets used by customers to upgrade older equipment to use newer code releases.

16.5. Manufacturing maintains a record of each device, including the code revision that was shipped inside it.

17. These process controls are designed to ensure that the code labeled for release on a certain device is in fact the code that ships with that device, and they do so with a high level of assurance.

[illegible]
Mario D. Santana

SWORN TO before me this 28 day of October, 2008.

[illegible]
Notary Public

MARIHAYDEE BORIA
NOTARY PUBLIC
COMMONWEALTH OF VIRGINIA
MY COMMISSION EXPIRES SEPT. 30, 2011
COMMISSION # 7130311

## Appendix A: Curriculum Vitae

Mario D. Santana, CISSP, CISA

Mario D. Santana joined the Secure Information Services group at Terremark Worldwide, Inc. in January 2006. He consults with Terremark clients on topics of security, technology, and risk management, and develops related consultancy product offerings. Formerly, Mr. Santana founded an identity management technology company and consulted for SteelCloud, Inc.

Mr. Santana has worked with numerous Fortune 1000 organizations worldwide, including financial, health-care and educational institutions, airport security and airlines, retail conglomerates, and technology and legal firms. He has led and managed engagements around security and risk management concerns such as corporate governance, forensics and electronic discovery, intellectual property fraud, insider incidents, and penetration testing and auditing networks, systems and applications.

### Recent Professional Experience

- Mario led the incident response team when a national financial institution was the victim of system compromise and subsequent internet identity theft fraud. The forensic evidence led to an investigation that spanned three continents and numerous intermediaries, concluding in containment, system recovery, root cause determination, and eradication of the breach.
- During a comprehensive insider threat assessment for a major provider of airport security, Mario found a fundamental issue of corporate governance and inter-departmental cooperation, after a full forensic investigation of suspected bad actors verified good faith and an excellent work ethic.
- A large car rental company was suffering system outages and severe monetary losses during an extended denial of service attack. Using a variety of techniques, especially digital forensics and reverse-engineering, Mario was able to pinpoint the root cause of the

weakness, and lead a team in the design and implementation of immediate work-around to bring the systems online while the database vendor developed a patch.

## Education and Certifications

- Colorado Technical University, B.S. Business Administration
- Certified Information Systems Security Professional (CISSP)
- Certified Information Systems Auditor (CISA)

## Professional Affiliations

- Member, International Information Systems Security Certification Consortium (ISC$^2$)
- Member, Information Systems Audit and Control Association (ISACA)
- Member, International Systems Security Association (ISSA)
- Member, SysAdmin Audit and Network Security (SANS) Institute and board of directors
- Member, FBI InfraGard, Dallas Chapter