

Weakest Link in Information System Security

Charles W. Flink II
cwf@infosecana.com

Snake Oil: *n.* - a western American 19th Century colloquialism referring to a fake cure. Thought to originally be a mixture of cheap alcohol and cod-liver oil sold by self-declared experts to trusting teetotalers at exorbitant prices. The customers were cleaned out (both wallet and otherwise) while being made to “feel good”. Known side effects: morning hangover followed by likely addiction.

ABSTRACT

Some claim that the 30+ years of failure in solving the puzzle of Information System Security (ISS) can be traced to a failure to apply time-proven Engineering Principles to the problem. This view questions the skills of the technologists involved and the quality of the “science” taught in the Computer Science departments of our institutions of higher learning. Others argue that this problem is a matter of corporate greed and lack of integrity. Still others argue that all will be solved by peer-review as a natural consequence of the Open Source movement. This paper argues, however, that the sad state of the ISS market is the natural consequence of high level of ignorance and paranoia about the “illness” on the part of both the *patient* and *doctor*! In short, we’ve been buying and selling *snake oil* in an effort to merely *feel good* at having done something to address a terrible disease we do not fully understand.

INTRODUCTION

The argument is made here that the root cause for 30+ years of failure in the Information System Security market derives from a failure to appreciate one of the most basic principles of security: *no security solution is ultimately stronger than its weakest link*. This principle, when coupled with human nature and the natural dynamics of the marketplace, has resulted in reducing many of us in the industry, myself often included, to the position of selling what prove to be “fake cures” rather than real solutions.

It is not that this principle is not well known. This *Weakest Link* principle dates back into antiquity, taking its name from the chain-smith’s shop: no chain is stronger than the weakest link in the chain. Certainly the principle itself goes back long before the making of chains. It was obvious to the earliest humans that the security of their settlement was no greater than the security of the weakest point in the defensive perimeter around the settlement. This principle is so basic as probably to be key to why our ancestors lived in caves! So how can something so simple and clear to thousands of generations of

Weakest Link in ISS

C. W. Flink

“security engineers” before us lead to turning well educated, dedicated, very “high tech” people into *quacks* selling *snake oil*?

I’m sure, at this point, you’re saying, “How *dare* he!” Let me assure you that I am not insulting anyone, nor exempting myself from any blame, if blame need be spread about in this matter. I hope by time you reach the end of this paper, you too will have come to share my opinion about our common history and how we can begin to move beyond it.

I’d note, before moving on, that the snake-oil salesmen of the 19th century were, in part, the fathers of the modern pharmaceutical industry. And don’t forget that the world’s most popular soft drink derives its name from a patent medicine of the 19th century that replaced the alcohol in snake oil with even more addictive cocaine! So though you may not appreciate my description of the “state of our art” as equivalent to quacks selling snake oil, please keep reading. Those quacks educated the consumer and themselves, ultimately striking upon worthwhile formulations.

Humor aside, there *is* a great deal of progress yet to made before we achieve professionalism on a scale appropriate to the need. Let me assure you there *are* high quality ISS products in the market and the vast majority of all the security engineers and engineering companies have very high integrity. It is the very *human* nature of human nature that allows bad results to come from well-intentioned people. And, of course, there *are* more than a few bad apples out there!

FUNDAMENTAL MISCONCEPTIONS

My first encounter with our customers’ misconceptions about ISS came shortly after demonstrating one of the earliest prototypes of System V/MLS.¹ I had designed Sec-Pac (or “Security Package”, as it was known then) as an add-on for System V to achieve a highly practical and supportable solution to the security problems inherent in contemporary OS design. Though evaluation under the TCSEC standard was the ultimate goal, I considered it my immediate goal, *as a responsible engineer*, to solve the underlying problem, not to simply meet the “letter of the law” as written down in the Criteria and referenced in the RFP.

The product was designed to maximize efficiency, portability, compatibility and supportability. The “KISS” rule was essential to success. There were very few of us, a small budget, and there were many, many versions of UNIX out there. AT&T Federal Systems did not even control AT&T’s UNIX and sometimes was forced to bid UNIX systems from other vendors to include the features required by government RFPs. As a result, Sec-Pac was designed as a pair of pseudo-device drivers that would integrate with virtually any UNIX kernel and “hook” into relatively few security relevant procedures

¹ AT&T System V/MLS was an extension of UNIX System V to include auditing, mandatory access control and assurances sufficient in 1989 to earn a formal B1 rating (with B2 features) from the NCSC under the Trusted Computer System Evaluation Criteria, a.k.a. “Orange Book”. System V/MLS was the first secure UNIX system to achieve a B division rating (“Multi-Level Secure”) under the criteria.

Weakest Link in ISS

C. W. Flink

within the kernel. One module provided the primary mechanism supporting auditing and the other provided the (optional) B-Division (labeled data) Mandatory Access Control policy. My philosophy was: “The only surely flawless line of code is one we can avoid having to write through the proper application of careful design and analysis.”

The result was an extremely compact product, with less than 100 lines of documented C-code in the MAC pseudo-device, less than 1000 lines in the SAT (Security Audit Trail) module, and well less than 10,000 lines in the various utilities and application-level components necessary for the management of security labels and analysis of audit trails. A large amount of effort, however, went into the analysis of the base System V code and the justification of the placement of the “hooks and probes” that coupled these devices to the kernel. The documentation, analysis and defense consumed over 80% of the 20 staff-years (over 2 calendar years) devoted to the development and evaluation of the system.²

Our product was an order-of-magnitude (or more) efficient in audit overhead than competitors systems when performing “full” audits as a result of a design that treated the system as a state-machine. We only recorded security relevant “transitions” and synthesized all other audit-event data from prior audit events, vastly reducing the size of audit records and the cost of recording them. Security labels were “overloaded” on the existing UNIX group ID data structures, allowing us to make *no changes what-so-ever* in UNIX data structures such as I-nodes, process tables, file tables, backup files, copy utilities, etc. Though structurally an “add-on”, logically System V/MLS was a clean integration of the concept of Mandatory Access Control into the UNIX security model.

Over the next 5 years, AT&T licensed System V/MLS to over a dozen other computer manufacturers, providing the basis for evaluated C2 and B1 systems from Harris, Amdahl and UNISYS, among others. We ported (or assisted in porting) System V/MLS to PCs based on the Intel 386 chipset, as well as tracking the evolution of AT&T UNIX through System III, System V and ultimately System V.2. The various MLS products died out as AT&T slowly withdrew from the terminal, computer and ultimately, the UNIX software markets. The progression was similar for our licensees as well as the whole “Orange Book” initiative.

So if it was so great, why is it now dead?

This returns me to that early day when I was demonstrating Sec-Pac to one of the first government visitors to our lab. If I had fully appreciated the implications of the “disconnect” experienced on that day, I would have (or hope I would have) gone to my management and reported that we were on a “fools errand”: the Orange Book was doomed, System V/MLS was doomed, and the subsequent 20 years of ISS failure was predictable.

² Another 5 staff-years were devoted to developing an MLS module for the AT&T DMD-630 windowing terminal. The evaluated system (TOE) included this windowing system providing classification labels on windows along with mandatory-policy mediated and audited “cut and paste”. The 630/MLS terminal was the 1st B-Division rated windowing workstation.

Weakest Link in ISS

C. W. Flink

System V/MLS added MAC policy and accountability through auditing to all “objects” extant in the context of the UNIX System V Application Programming Interface (API). In other words, access was controlled and audited to files, memory segments, processes, sockets (ultimately), streams, devices, ports, etc. If a UNIX administrator or systems programmer dealt with it, we managed and audited it.

To prove the value of securing an operating system, I selected a then popular “office desktop” (email, editor, file manager, etc. with menu interface and scripting language) that cleanly exploited the UNIX objects. Each email message and each document was mapped into a file. Each “desktop folder” was a separate UNIX folder; users were users, etc. The menu system was implemented via a scripting language. I was able, in a matter of a day, to add the concept of “classification level” to the menu.

I was able to present our guest with a working office platform supporting Multi-Level Security, including classified email, classified documents, etc., all neatly partitioned into their own mandatory compartments. A simple interface allowed the user to login and logout in order to transition between levels and compartments as necessary. It was clearly a tour-de-force with which I was quite proud. It cleanly prevented the compromise of information via Trojan-Horse attacks and allowed a complete office system to be used *safely* without need for modification to the application code, trusting of the code, or even *analysis* of the code!

Inverted Pyramid Principle - Design to minimize the amount of trusted code; the bulk of the software should not be entrusted with critical security functionality! The security of the total system must derive from a crystal clear, diamond hard pinnacle of trusted code upon which the bulk of the system rests.

My guest immediately suggested that it was nice, but he wanted the folders to list all documents, regardless of classification. And the mail system should list all messages, with the classification neatly shown beside the subject line. And the editor was ok as word processors go, but he wanted the classifications to be on a paragraph-by-paragraph basis, not just one classification for the whole document.

Obviously, the entire concept of risk-minimization embodied in the Inverted Pyramid Principle meant nothing to this customer! What he wanted was a privileged and trusted *application* requiring vastly more trusted code than we could *honestly* support. Now, the sales force immediately stepped in and assured the customer that we certainly could provide those “minor” enhancements, “If Flink can secure UNIX, then certainly he’ll have no problem securing a simple application!” My objections fell on deaf ears.

We’d been baited and switched! The authors of the TCSEC, quite unintentionally, “sold us a bill of goods”! They told us if we were to develop a secure operating system, they (the government) would buy the system. They told us *security is critical*, it is more important than mere *features*. Unfortunately, they didn’t explain this to the very users who were going to *pay* for the system. In a strained sense of “fair play”, one flask of snake oil deserves another, so we (along with the rest of the industry) succumbed to

Weakest Link in ISS

C. W. Flink

selling a promise of *future* secure applications to a customer barely willing to pay for a secure kernel!

The history from here is well known. The “Red Book” extended the interpretation of the TCSEC into the evaluation of secure networks. Another extension was promised to address the evaluation of secure databases. The chain of exchanged promises hyped-on for a few more years. Finally, it was realized that neither side could deliver on their promises. The customers willing to sacrifice ease-of-use for security simply did not exist. The products build to the standards of assurance embodied in the TCSEC (and later in the ITSEC) were simply not going to be built.

The final nail in this coffin was struck by the development of the PC. Federal Systems made far more profit reselling Taiwanese personal computers listed under our GSA contract as “graphics terminals” than selling our own UNIX systems! The customers wanted PCs and knew how to manipulate the procurement system to get them.

Most striking from the perspective of the security engineer, the PC embodied *complete rejection* of the entire suite of security technologies developed over prior years: no login, no user identity, no access control, no auditing, no privilege restrictions, no authority!

The government’s thrust to promote improved security surely did seem to convince their own users of one thing: *they didn’t want this type of security!*

Note: I didn’t say they didn’t want *any* security. In fact, the physical security of the PC is attractive at the local level: there is no bureaucracy that has to be trusted to make your system secure; the guy at the gate with the gun sufficed! Likewise, you could easily walk up to anyone’s PC and see what he/she was doing. Nothing was hidden; nothing was complex. Nothing was “outside the box.”

The World Wide Web is *another* example of “security” taking a back seat to utility! The Internet long predated the WWW (e.g. ARPAnet). The revolution was delayed until an obscure CERN programmer proposed standardizing file-browsing techniques across R&D facilities to make it easier for the nuclear scientists that were his “customers”. No security was “built-in”, the software was public domain and the whole idea was free and easy sharing. The result was an explosion of use and a revolution in the way research, education, business, government and the military work from day to day. The only bow to security was the development of SSL, the most classic example of a “retrofit” security solution since, since... SV/MLS?!

The daily news is full of example after example of ease-of-use and utility winning out over “security technology” in market after market. But worse, in spite of the revolution in the use of personal computers and networks, the greatest *real-world* security failures in the history of the CIA, FBI and DoD can be traced NOT to failure of security technology, but to the failure of *human beings* to resist greed, envy, sex, power and/or laziness. Simply review for a moment the panoply of spy cases over the past decade.

LESSONS LEARNED

So what did I learn? *Brilliant people are among the dumbest on earth.*

By “brilliant”, I mean people who are facile with abstractions, undaunted by complexity and generally able to leap to the top of any IQ test with a single bound. By “dumbest” I mean likely to make the wrong decision whenever faced with the realities of life and work. If you’ve managed programmers or mathematicians, you know what I mean.

When working as a team leader in the systems programming branch of a DoD computer center, I once had the opportunity to supervise a summer co-op who was a classic example of this phenomenon. He had scored perfect on his SAT, won a full ride to his state’s most prestigious university, had been honored by his Governor as a brilliant example of what the mid-west could produce. And all of this was before his 16th birthday!

This young man strode in with a calm confidence that verged on becoming a sneer. Our department head was showing him around, introducing him to us first since we were the “research group” and, in the mind of our DH, was least likely to reveal ourselves as an embarrassment. Our new branch head came in late, having been off on some errand for the DH, and began nervously glad-handing his way into the introductions.

“Welcome aboard, Steve! Sure glad to have you with us. I hear you’re from Columbus, Ohio.... I’m from there too!I just found out.” Our BH was dishing typical BS to cover his being late to the introductions. Steve, calmly and coolly replied, “You just found out you’re from Columbus?” highlighting the minor order inversion. It was a devastating put-down, delivered swiftly and surely and without effort, as if his opponent wasn’t even worth expending the energy of adding a sarcastic tone to the remark. Brilliant! ...but a remarkably dumb way to introduce yourself to your boss!

It did, however, immediately bond us young firebrands with Steve. We immediately adopted him as the “younger brother” we all wish we had! Unfortunately, in retrospect, we were too young, ourselves, to appreciate how much more than “brilliance” was required in solving real-world problems. We failed to teach the humility, discipline and respect for elders this young man needed. With certainty, however, life by now has completed teaching these lessons.

The people who pointed out the dangers of Trojan Horses, invented the concept of MAC and wrote the Orange Book were doubtlessly brilliant. Unfortunately, we’ve all been proven dumb in our ability to predict the future of our technology and dismal failures in appreciating the dynamics of human society.

The good think about brilliant people is they *can* learn, once they overcome their prejudices!

Weakest Link in ISS

C. W. Flink

[Bruce Schneier](#) relates a similar “awakening” in his evolution from a crypto-hawk to a much more mature security engineer today. Bruce relates this in the preface to his most recent book, [Secrets and Lies](#). Charles Mann writes brilliantly of this phenomenon in his Atlantic Weekly article, [Homeland Insecurity](#). Hopefully, we’re now ready to learn these lessons and enter an era where information security can empower the individual, corporation, and government rather than remain a drag on progress.

The sooner we embrace *positive*, socially and technically *integrated* approaches to security, the sooner we’ll see an end to the long dark history of failure in our profession.

NEXT STEPS

Where do we go from here? The first 5 pages of this article were largely written over 2 months ago. I’ve waited most of the summer, waiting for insight on how to end. I’m already over the goal of 5 pages, and could easily provide examples expanding this into several chapters of *introduction* to a book! We’ll have to save the bulk of the discussion for the workshop in November. But I will provide this “outline” to finish the page:

Security is a holistic endeavor. It encompasses far more than the physics of locks, the mathematics of ciphers, the analysis of risk and the tactics of the military. It includes the psychology of the citizen (computer user) as well as the mind of the attacker (terrorist?)

Security is a never-ending battle. Ah, the [Yin-Yang](#) of it all! For every action, a counter; for every good, a balancing evil. Accepting this reality and judging solutions as achieving an ever changing, dynamic balance is essential to avoiding the pitfalls of absolutism. ...who wants to be in a rut anyway?

Accountability out-weighs confidentiality. We’ve spent a great deal of time and energy trying to prevent the possible-but-not-yet-imagined by protecting secrets. Far more practical and effective would be detailed logs of all security relevant events in our systems, information and otherwise.

For too many years, we’ve attempted to skip the “core” of the solution. We’ve engineered Information Technology Security, and then jumped immediately to marketing it. We left out fundamental parts of the holistic solution:

- Engineer ITS
- Legislate ITS - core problems
- Litigate ITS - core problems
- Insure ITS - core marker of progress
- Market ITS

When we see users and industry accepting accountability, society formulating laws, courts testing those laws and insurance companies mitigating failure, *then* we’ll know we have a product *worth* marketing!