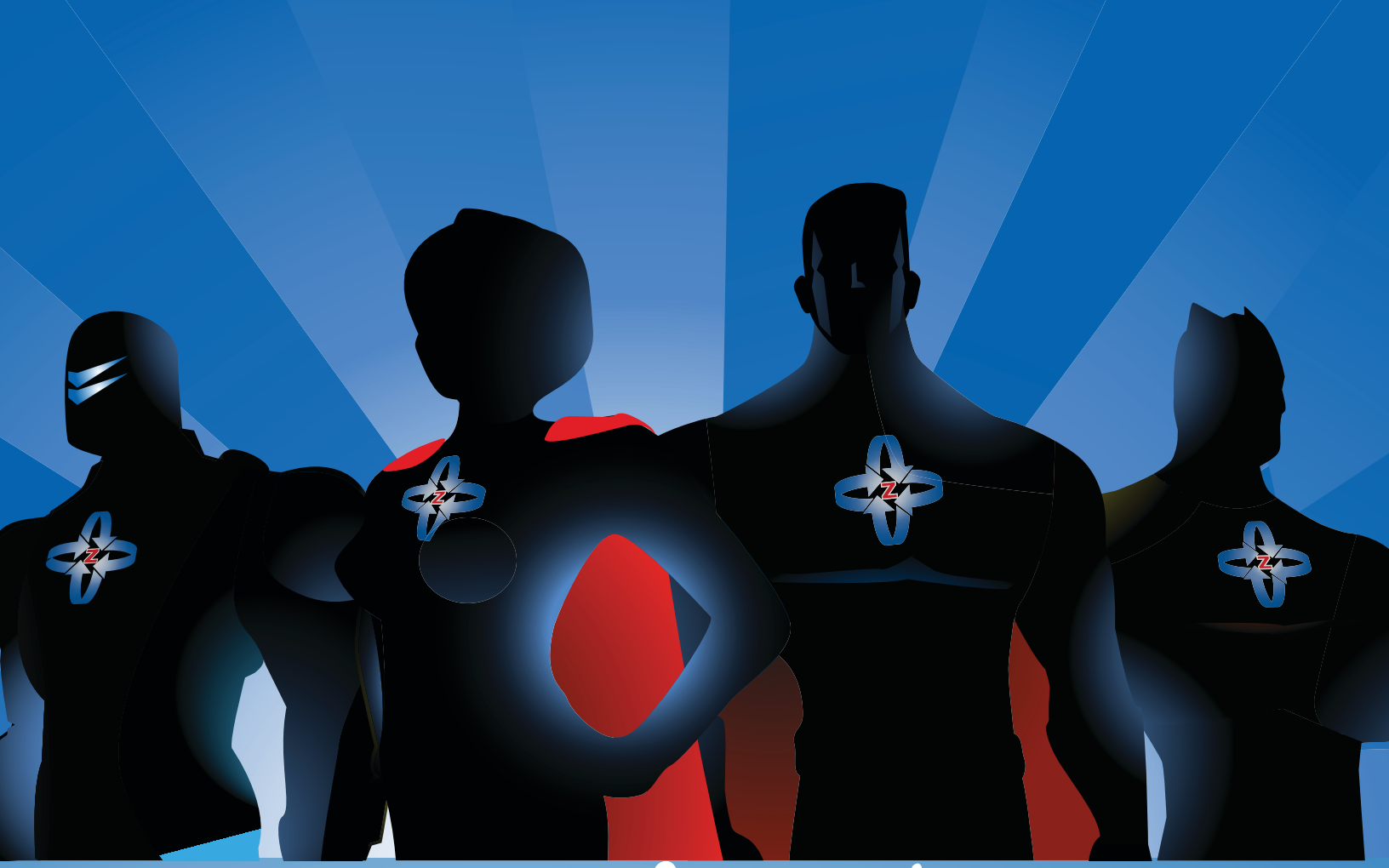


THE ZEXCHANGE

Introduction

Once upon a time in a galaxy not so different to our own, a phone conversation took place. It was only scheduled for 30 minutes but some hour and a half later a plan had been hatched :-)

That conversation was between little old me and the Head Geek in charge at New Era Software, Paul Robichaux. It turns out we'd both been worrying about the same thing! The fact remains that most mainframe technicians are heading rapidly towards retirement and there doesn't seem to be generations of successors lining up to take over.

Some organizations have been trying to train up the next generation over the years. But it's been hard to generate interest in a field that students think is dead end technology. Did you know that IT is the only industry that treats the word "Legacy" as a bad thing?

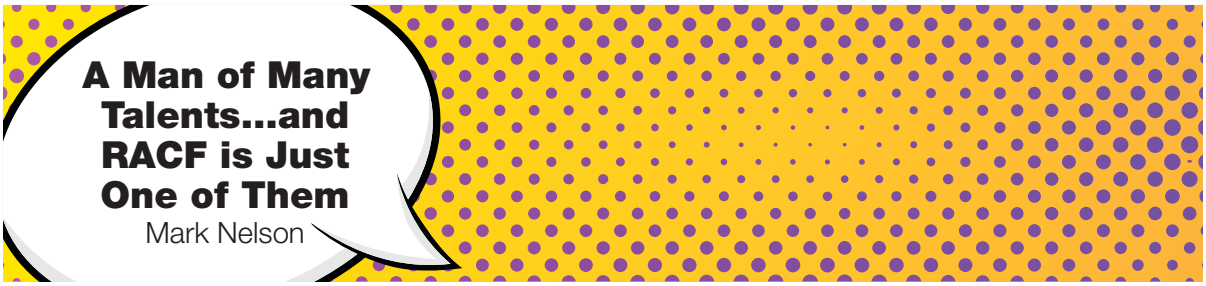
Small groups have been springing up around the world where the mainframe is celebrated! (see, for example: http://millennialmainframer.com/2015/11/connors-personal-mainframe/ and https://www.youtube.com/watch?v=45X4VP8CGtk). We need to get over that idea that kids don't want to work on green screen. That's old hat! Anyone who consciously chooses "vi" as their editor certainly isn't going to be worried about ISPF being old fashioned. And Linux is run with a command line interface. This is not old technology any more, it's rebranding as "Classic" and we're getting a whole new audience because of it.

But Paul and I agreed that the rate of pick-up has not been quick enough to date and we have to have a backup plan. So that was when the idea of documenting many of the more complex parts of mainframe operations came from and (here was the "radical" part) making it freely available to anyone who expressed an interest in mainframes.

And so the z/Essentials series of publications was born ☺

We've gone on to document the lives of some of the most influential men and women who helped to define the mainframe environments we run today. These folks are the "Superheroes" in this industry. They are the ones who allow us to keep the money moving and the power and water running. The ones who made it possible for the international travel that we enjoy today and even plunge right into the Internet-of-Things, managing back-end electrical power for the internet-attached, Nissan Leaf motor vehicle! (see IBM zIQ -- Can you go five minutes without using a mainframe?: https://www.youtube.com/watch?v=afwcIBj80hs).

What we're doing here is putting together the biographies here in one place. It feels right to be celebrating my "Superheroes" in this way. But I'm secretly hoping there will be more in the series to come.



Foreword

When I was asked to write this intro, my first thoughts were about Mark's strengths and why I consider him such a good friend. He's extremely knowledgeable. He's multi-talented. And he's just a nice guy. And that last one is the most important.

One of my earliest recollections of Mark is attending a SHARE session on DB2 security, a topic that I knew nothing about and needed desperately. That was the first time I saw him passing out (or in some cases throwing across the room) candy as a reward for questions and interactions. Since then, I've heard a complaint that his many 'Best Speaker' awards are because he has bribed his audience. But that's so not true. Mark is a great speaker not for his candy, but for his knowledge and interaction with his audience. Mark knows his stuff and then he is able to communicate that knowledge and finally he makes the whole process fun and entertaining.

While he is a true IBMer, he's not afraid to take a jab at the organization. Some of the parodies that Julie-Ann mentions have quite a few fans within IBM. And that irreverence occasionally pops up in his sessions too. He's not afraid to poke fun at IBM or himself.

Shortly after retiring from IBM, I had the opportunity to speak at the GSE conference in the UK. Not being a very experienced international traveler, I asked Mark if I could tag along with him. He graciously agreed but with the warning that he wanted to spend a couple of extra days just touring. Because of Mark, I got to visit Bletchley Park (a thrill for a crypto guy), St. Paul's Cathedral and attend my first, and only, West End Theater show. What a diverse set of experiences! And Mark was my knowledgeable guide for all of them. Mark did all the driving on that trip and I was supposed to be the navigator. We only got hung up in the roundabouts a couple of times.

In addition, Mark decided that he was overdue for some flying time and decided that he would rent a plane to get an hour of pilot time. There aren't many Americans that can say that they've been flying in a small plane over the English countryside! But I'm one because Mark took me for a ride.

I've been privileged to call him a coworker and honored to call him a friend.

Greg Boyd
MainframeCrypto

A Man of Many Talents... and RACF is Just One of Them

This is the long awaited fourth paper introducing a "Who's Who" of some of the brightest, and yet under-sung stars of our mainframe generation.

How I always think of the ever youthful Mark Nelson

It's starting to sound insincere for me to keep talking about how lucky I am to be able to work with so many of my heroes. So, for now, all I will say is that whenever I hear Mark Nelson's name I immediately start trying to think of some complicated RACF questions so that I can win chocolate! Whenever Mark presents at user groups he brings huge bags of Hershey chocolates to be passed to (more accurately described as launched at) anyone who asks any question.

It turns out that this fabulous ice breaker technique, which gets even the most technically minded folks talking in front of other people, was shamelessly "stolen" from Cliff Stoll (Yes, he of "The Cuckoo's Egg" fame - the book that led me to want to learn how to hack!). One story, two heroes, I say again, I am very lucky in my chosen career!

I met Mark at a GSE (GUIDE SHARE Europe) conference many years ago and he struck me as the antithesis of a "normal" systems software developer. Significantly younger than most of the other IBMers, he was obviously very clever, being responsible for design and development on some of the new (some might say best) bits of RACF, one of my favorite software products. But he was also approachable and always interested to find out what the customers are actually doing with his "baby", RACF. I've known Mark to work really closely with a single customer to make sure that a change leaves them with exactly what was wanted/needed.

I would describe Mark as a bundle of fun! Just getting enough detail to write this short piece was a challenge as we kept veering off to talk about completely unrelated subjects dear to both of our hearts!

Mark's early life was spent in the heart of a very close family in Queens Village, New York City and family is very much a theme that runs through Mark's life. Whether it's his actual family or his work family doesn't seem to make that much difference to one of the only people I know who can always see a positive in any situation. He describes an idyllic childhood living with his Mom (who worked as an executive secretary until her babies started to arrive) and Dad (a former professional baseball player who worked in the insurance industry) as well as 3 brothers.

Mark sent me the following message after seeing the first draft of the article with my selection of images and I enjoyed it so much I wanted to include it:

"The photo above is more appropriate than you might think. That's the Unisphere at Flushing

Meadows Park in Flushing Queens. My very, very first “real” job was working for the company that had the food concession for the park. One of the places that I worked was a little food kiosk that was at the base of the Unisphere. If it still stood, it would be between the two towers in the background (The New York State Pavilion from the World’s Fair in 1964, made famous in the movie “Men in Black”. The building that is just out of the photo to the right is the City Building which housed an ice skating rink that I worked at for five years. Everyone one should work in fast food at some point in their life! ☺)”

Another quote from Mark about his start in life that I particularly liked was “Six people, one house, and one bathroom”. I can only imagine the fun and games in that household in the mornings! And, with so many more of his close family living in the immediate locale too, it’s easy to understand why Mark always felt supported. It’s so nice to talk to people who feel blessed by their start in life. ☺

I’ve spent most of my career working with people who didn’t want others to know their age (or ANY personal details) and, in this increasingly security conscious time, today it actually makes sense not to probe. Particularly if the subject under discussion has a significant web presence and presents themselves to the world as any kind of “IT Security Expert”. Let’s just say that growing up in Queens in the 1960s brought its own special set of challenges.

Mark takes up the story:

“I was fortunate to be granted a seat in an all-scholarship high school in Manhattan, NYC. While the school was a one hour and fifteen minutes trip (one bus and three subways) each way, it allowed me to explore and enjoy all that Manhattan had to offer.”

The incomparable Fats Waller

His passion for music really took off at high school and is something that still consumes him today. He’s gone from being woken up with the sounds of his, very talented, Mom playing Irving Berlin and Fats Waller on the family piano to running a very well respected choral society as part of his extended “duties” at IBM. The Mid-Hudson Valley IBM Chorus had been formed at the personal request of former IBM CEO Thomas Watson Sr., and Mark joined in the early 80s becoming its director in the late 90s.

His musical achievements include that first high school appearance in the chorus of “Paint Your Wagon” right through to taking piano lessons alongside his daughter in 2005. The pair went on to perform duets at recitals together.

But I digress… We were talking about Mark’s journey through life and I got distracted by one of his other passions! Something that has been really quite hard to avoid on this particular exercise! :-o

He says that his choice of college was primarily based on locality and, given the arduous journey he had to get to high school, I can see why! He chose the Polytechnic Institute of New York (which used to be known as the Brooklyn Polytechnic - before it morphed into the Polytechnic Institute of New York, then Polytechnic University of New York (Go PUNY!), then Polytechnic Institute of New York University, New York University Polytechnic School of Engineering, and now New York University Tandon School of Engineering - Mark says: “It seems to have had as many name changes as MVS has had!”).

He started as an Electrical Engineering (EE) major. But in the spring semester of his first year, Mark had to take his first programming class. I’ll let him explain:

“While I had done some rudimentary programming on my trusty HP-25 calculator, I'd never touched a "real" computer before. So before I started the class, I got a manual for the language that we were using (a variant of PL/I called PLAGO, for "Polytechnic Load and Go") and a deck of blank punch cards. From the very first sample in the book that I punched, submitted, and got the output for, I was hooked. I changed my major to computer science and never looked back!

The computer science curriculum was tightly integrated with the EE department, so we did a lot of geeky things like bread-boarding our own circuits, writing microcode, and writing assembly-language programs, and even our own primitive assembler. I liked what I did and managed to get a job working in the computer center, where I discovered the joys of being an operator on an RJE station for a System/360 Model 65.

IMHO, everyone in IT should work as an Operator at least once."

It was during his time at college that Mark met, and fell in love with, his significant other, Julie. He says: "Getting an education was my intention in college. Finding a soul mate was the best side benefit ever!"

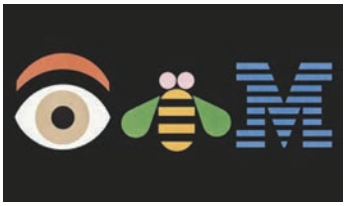
He's an old romantic at heart so, during a family vacation to Hawaii he managed to arrange for them to renew their vows under the tropical sun to celebrate 25 years of marriage. This fact was kept secret from Julie despite so many people being involved in the occasion. I can't help but admire this man who goes to such lengths to remind his wife of how much she means to him. ☺

Mark and Julie after renewing their vows on a beach!

I overheard Julie talking to someone at a technical conference in the UK recently and it was completely apparent that the feeling is entirely mutual. She was explaining that she didn't always understand what Mark was talking about at these events as she's "non-technical" but she can see from the audience reactions how highly regarded he is. Can't say that I heard anything I'd disagree with! (P.S. Sorry Julie, I promise I wasn't stalking you! ☺)

Mark's first "real" job in computing saw him working weekends as a Computer Operator for a company called Time Sharing Resources (TSR). He was supporting 2 System/360 Model 75 mainframes (each with a whole 1Mb of storage!). The systems supported several hundred time sharing users with sub-second response times.

After two years at TSR, he moved on to a part-time co-op job at Grumman Data Systems (GDS) as a VTAM Systems Programmer. GDS had one of the largest data centers in the north east, supporting Grumman's extensive manufacturing activities on Long Island. This top-notch team took Mark "under their wing" and expanded his mainframe horizons. He thoroughly enjoyed his time there and says: "I spent two years at GDS and more than 34 years later, I'm still in contact with these wonderful folks."



Cool IBM logo from the 1980s

College graduation beckoned for Mark in 1982. IBM had been recruiting at Polytech for years and the interviewer always had a "full dance card". At the last minute, IBM decided to send 2 interviewers. The only problem was that there were no students to speak with the 2nd interviewer. The night before, 1 of the administrators at Poly asked Mark to sign up with the 2nd interviewer just to make sure that there was someone for him to talk to. He did, and managed to get 2 interviews at IBM in Poughkeepsie! He was offered and accepted a job working in a group that was doing database design, modelling and system configuration.

And so the next chapter begins...

Interestingly, the database job came in the wake of the experience Mark had of writing a VTAM Systems Configuration Management tool whilst working at Grumman. The job he took at IBM saw him covering some of the same configuration management topics but in a database environment rather than out on the network.

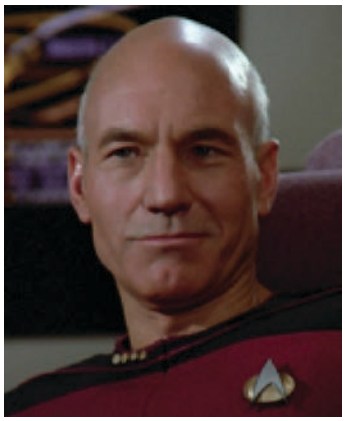
The product that Mark had been hired to work on never actually made it onto the market but what the whole experience did do was expose him to the IBM process. It also taught him some valuable lessons about how goals sometimes evolve or even completely change over the duration of a project.

When the database job was over, Mark (along with the rest of the team) was asked to take a temporary, 6-month job in the RACF development team supporting the implementation of B1 capability aka MLS. I love this comment from Mark about this latest twist:

“In a way, working on security brings me into the family business: Two of my brothers are retired New York Police Department (NYPD) officers and the other is a Battalion Chief for the Fire Department of New York (FDNY). We've all worked at making the world a safer place.”

Mark’s first assignment was to work on the RACF Data Set convert Utility, which converted the RACF data set from the old, non-restructured data format to the new, restructured format. The overall (high-level) design was set by Walt Farrell. Mark’s first job was to make Walt’s design a reality. Walt Farrell, for those who aren’t aware of him, is another of the deities of the RACF World.

Working on that utility gave him the background to create the RACF Database Unload utility, which Mark designed and developed from the ground up. He used the code, which is used to create a RACF database offload, as the coding project for his Master’s degree! This popular feature allows the security data to be processed either in raw form or in DB2 tables which can be optimized for easy of querying and reporting performance.



The utility has been a huge success and he still gets as excited talking about it today as he did when he asked for permission to execute the project and got an email back from his manager, Rich Guski, simply saying **“Make it so!”** ☺

Perhaps one of the most significant tasks he’s undertaken is the digital certificate processing which RACF is capable of. Again, Mark worked as part of a larger team to implement someone else’s design (credit for which goes to Jim Sweeny).

“The work is very interesting, the team is fabulous, and I've managed to stretch this six-month assignment into 29 years. We're really a tight knit team. We've got some lovely traditions of our own. My personal favorite is the annual holiday pot-luck lunch which features the RACF and Friends Holiday Singers which for the past twenty-six years has been affectionately "biting the hand that feeds us" by writing and performing parody songs that highlight the best and the worst of the prior year. We have written and performed over two hundred songs for the RACF team.”

Mark tells me that he is finding it fascinating to be working in an environment where many of the developers under his wing weren’t even born when the RACF code was first written! But I think that’s a challenge for a lot of us “grey beards” - Doctors, Policemen and Software Developers are all starting to look like teenagers! But it’s these “teenagers” who are going to be supporting our platform of choice for the next 50 years of its existence so we must get involved.

Mark teaches online classes at Marist College and, despite yearning for a time when we did education face to face, is busy making a difference in these youngsters’ lives.

He’s not often on the road but when he is, it’s often presenting to IBM user groups. Mark traces back his joy in doing this to an early presentation he was encouraged to give, at a small event that IBM were doing internally for RACF’s 15th anniversary in 1991, by his Manager at the time, Frank Witham. Mark takes up the tale:

“Eldon Worley (known in the industry as “The Father of RACF”), whom I had met at SHARE a few years earlier, generously provided me a majority of that information and gave me a list of people to talk to for additional information. I’ve spent the past twenty-five years adding to that presentation, culminating in a joint presentation with Eldon at the Vanguard Enterprise Security Expo last November.”

The tasks assigned to Mark (along with the ones he picked for himself) all led to him gaining a massively detailed understanding of the internals of RACF. And that’s the real reason he’s been in the job for 29 years with no sign of any imminent change in circumstances. As customers do more and more wacky things

with mainframes, RACF must keep up, and Mark is an integral part of the team that ensures this happens.

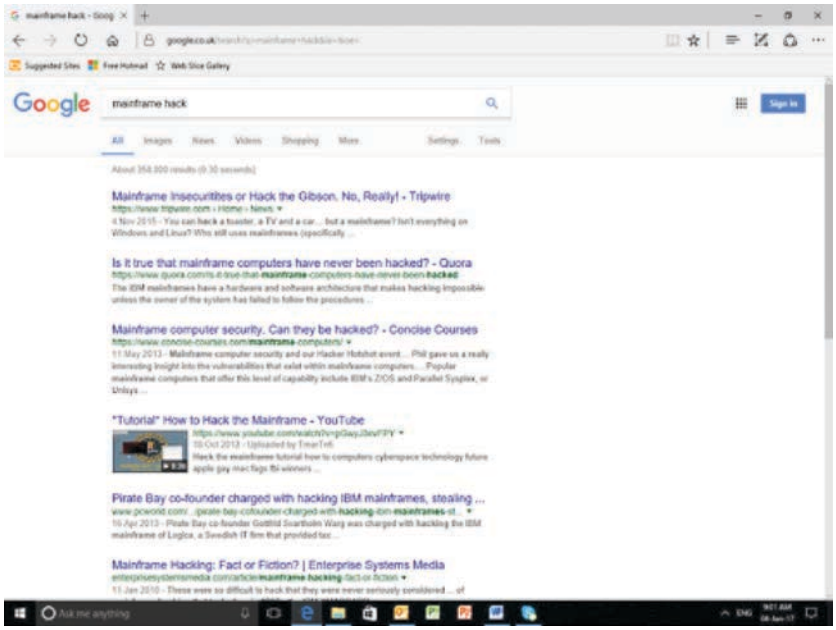
We RACF users salute you!

I asked Mark what he thought we'd be focusing on next in the mainframe security world and I think that I'd have to agree with his view that our focus really NEEDS to be on intelligently deploying what we already have available to us. I'd add to it personally by saying that the consultancy which I work for has seen a massive increase in organizations tackling the basics. And by the basics I really do mean effectively starting their RACF (and/or ACF2 and/or Top Secret) definitions almost from scratch!

All of those stories that we've been seeing in the global press about hacking incidents around the world are starting to have an impact on our day-to-day work. "Elevated access privileges" is a phrase we are all starting to see in our audit reports. And organizations are beginning to realize that security policies which were defined many decades ago are no longer fit for purpose. Sometimes this is because the business has changed shape significantly in the last 30-50 years. Other times it is related to security decisions which were made before the systems were internet connected!

This is an exciting time to be involved in mainframe security that's for sure. ☺

Mark told me about a related effect which used to be seen in the United States and seemed to have been caused by putting pictures of missing children on the back of milk cartons. Whilst any abducted child is a disaster of epic proportions, what this campaign did was associate the idea of child abduction with something as everyday as milk. Whereas the actual risk of a child being abducted was and remains, statistically, incredibly low. The result is was that the biggest fear of most children in America at the time was that they would be abducted!



He explained to me that: "As "Cybersecurity Professionals" it is our job to make sure that we raise awareness of what security **SHOULD** mean to our organizations. But we have to make sure we're not just causing panic! We need to work with appropriate partners (internal departments and/or external organizations) to properly profile the risk associated with **NOT** taking a new course with regards to security.

Legislation and regulation have always been a driving part of what we do. But, we must also understand what the current threats are and adapt our environments before we are a victim of those threats. In the 21st century we are seeing the prospect (or increasingly the actuality) of having embarrassing hacking stories associated with an organization being used to drive change. Bad corporate choices can, and DO, affect everyone!"

Attitudes are changing. And whilst it is very tempting to stick with the old "But we've ALWAYS done it like this" approach, it's vital to get on top of what your organization is doing across the business so that you always have a valuable opinion to offer. RACF offers a very rich toolset for helping to secure your organization's business and the mainframe offers a value for money platform for many new applications even if they aren't classic (DON'T CALL IT LEGACY!) mainframe workloads.

One of the areas Mark was eager to point out that z Systems can be very helpful with is encryption, with

features like CPACF and the CryptoExpress cards in the hardware and ICSF in z/OS. IBM recently issued a SoD (Statement of Direction) for z/OS which indicates their intention to augment the current encryption of data a rest (implemented in IBM's DS8K), with the encryption of data at the z/OS data set level. Exploiting this new functionality will enhance the security & privacy of your organization's data with the exceptional quality encryption that z Systems provides.

As Mark put it "Encryption is easy. Key management is the hard part!"

But does your organization even know about this or are they off trying to shoehorn some kind of PC based data centralization process into your business to enable these kinds of function? Don't you think this would be a good time to let someone know that we can do all of this for them on a platform that is already on the ground **AND** already contains the vast majority of the data?

Do your users know what security means to your organization? Your security is only ever going to be as good as the weakest link in the chain and if your users don't know that the Help Desk would never ask them for their password, are you confident that a good "Social Engineer" couldn't just talk their way into your systems?

Further into the future, predictions become trickier. The world of computer security is moving at such a pace it's almost impossible to say what will be bothering us in 5 years' time! As Mark said "Envisioning the threat event horizon is becoming increasingly difficult! But there are always going to be Good Guys and Bad Guys. And the Good Guys are always going to have to remain reactive to what the Bad Guys are doing."

That said, Mark's opinion is that we are likely to become more dependent on automated intelligence. Business processes are generally fairly rigid and predictable and make a great target for automation. New heuristic systems capabilities (those that learn to recognize what a normal running system looks like and flag abnormalities for action) may take a while for us to get to grips with but will save carbon-based life-forms (i.e. you and me - and other humans) for the more specialist tasks.

Another thing that is bothering him is the Internet of Things. That is, all of the internet-connected, smart-devices that are increasingly permeating our lives. From smart fridges that know when to order more milk through to automobiles managing all sorts of things remotely (such as the Nissan Leaf which actually does most of its battery management on a mainframe at the back end!), these devices have, at best, inadequate security. I have even seen manufacturers of such products try to defend the complete lack of security as something they "had to do" because there wasn't enough room to do all of the clever stuff they need to AND pay attention to security.

Quite frankly, it's not good enough. These devices have already been used by hackers to effectively shut the internet on October 22nd 2016. That attack saw hackers directing "bots" that had been installed on IoT devices to initiate a Distributed Denial of Service or DDoS against a company called Dyn (pronounced "dine"). Dyn is one of the organizations which provides the internet with name resolution so that when you type www.facebook.com you go to the right place. Dyn said of the attack (which impacted globally) it wasn't computers, it was "tens of millions" of Internet-connected things, like CCTV cameras, DVRs and routers."

And you don't need to be some evil hacker genius to get in on this type of action. A rudimentary search

using Google shows that anyone can order up the services of 10s of thousands of “bots” for as little as $50! The same rudimentary search technique can also be used to give you some ideas of what you could do with them!

Another of Mark’s mantras is “The greatest enemy of security is complacency”. Have a look and see what could be done to your systems and you might be more prepared to start from the same position as the NSA does. They always assume that they have already been breached.

So let’s stop being defensive about the way we’ve always done things and get on a new bandwagon. “The Threat Landscape” is changing and we need to keep up. The good news is that people like Mark and I see it as our job to help you make your organization understand what they’ve got and how to exploit it for the most benefit to your business.

The other good news is that we understand that some of your organizations might not yet be fully convinced that the 21st century applies to them. We can almost certainly help them to understand something else Mark told me: “The price of security is eternal vigilance”.

An example that Mark gave relates back to his hobby of flying. Like catching a hacker in the act, flying is governed by a large dose of skill backed up with an element of luck.

This story saw Mark taking to the skies in a single engine, light aircraft with his young nephew to watch a local fireworks display from a unique perspective. The pair decided to do a few turns around a point over Mark’s home to give the family chance to see them flying overhead.

This apparent indulgence also allowed Mark to gain altitude for the rest of the flight. I’ll let him take up the story:

“At SHARE and the Vanguard conference this year, I did a session comparing the aviation industry with the computing industry, specifically to see if we can extract some techniques to reduce defects (including operational errors) in information technology.

With regard to training, every pilot (even us lowly private pilots) has to train to a set of standards. Our skills are evaluated prior to being granted a pilot certificate. For my license, I have to get recurrent training every two years and demonstrate my proficiency to that set of standards. Normal and emergency procedures are both reviewed and demonstrated. Airline pilots do this far more frequently.

I can testify to the need for such training. Every year (weather permitting), I go flying on my birthday. Nothing major, usually just a little sightseeing or a short trip to a nearby airport to meet by brother and his family. One year, I was doing a pure sightseeing flight. I was going to do a little night flying to see a local fireworks show from the air. I decided to take my barely teenage nephew with me on the flight. We departed Dutchess County Airport and headed north-northwest to the Hudson River airport to meet my brother.

After a few minutes of flying, I heard a loud *bang* and the engine began to vibrate violently. I reduced power to stop the vibration, turned back towards the airport, and told the tower that I was returning because of a rough engine.

That's actually the short version. Since I had been through an engine failure scenario many, many times, without thinking, I lowered the nose of the aircraft to the "best glide" (most distance gained per foot of altitude lost). I determined that I had sufficient altitude to return to Dutchess County Airport and made my

turn.

After landing, we shut down the engine and pushed the aircraft off to the side. A member of my flying club stopped by and asked me a lot of questions: What RPM was the engine running at when it failed? To what RPM did you reduce the power to get the vibration to stop? What was your altitude when the engine failed?

I didn't know the answer to any of these. I was functioning on "automatic" as I had been trained to do. We need to make sure that we are trained to respond to threats, a response which has been both taught and tested. That testing (and a lot of luck!) contributed to a successful return to earth."

We cyber security professionals can learn from this story and not just that Mark's a pretty good guy to have around in a crisis!

While we might not always know exactly what shape our crisis (e.g. a cyber-attack) might take, we can be so ready for one happening that we can operate on "automatic", following well-rehearsed drills when such a thing occurs.

Less complacency and more vigilance will allow us to know where the possible entry points to our systems could be compromised. Test those boundaries so regularly that it becomes as normal as Mark's response to, what turned out to be, the catastrophic and complete failure of the only engine keeping him in the sky!

As Mark put it: "You don't know how well you have prepared until you test your preparation."

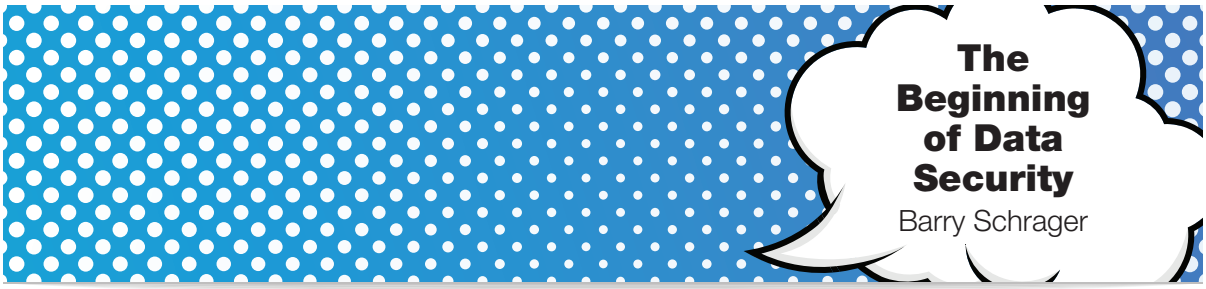
Mark does most things with all his heart! Whether it's singing in public, helping to run a Friends of Poughkeepsie Library book sale or flying himself around in his light aircraft, he is "all in".

It's this full-on, life-is-for-living approach that makes Mark such a magnetic personality. You can enjoy just being in the glow of this brain that runs at a million miles an hour. But don't try to get him to talk about himself… he'd rather find out about your life story than "be boastful"! ;-o

Working with Mark on this project has been a joy! Even if it didn't feel like work, more of a series of chats between old friends with similar interests, I'm going to count this amongst my favorite jobs, EVER!

Thanks Mark for being a delight and thank you all for reading ☺

Remember: Civilization still runs on MVS – Julie-Ann
(Shamelessly adapted from one of Bob Rogers's old tag lines)



Foreword

I entered the mainframe industry in the late 1980's when I began working for NaSPA, the National Systems Programmers Association, now called Network and Systems Professionals Association. To many of you reading this, I am a relative latecomer to the mainframe world. Yet, as publisher of Technical Support magazine then and with NewEra Software now, I have worked closely with authors, columnists, speakers, competitors, colleagues – many are practitioners and all subject matter experts, which is infinitely more important. I count myself as one lucky person to brush elbows with greatness.

As the Director of Strategic Partnerships for NewEra Software, Inc., I have noticed over the years that many of the individual contributions of the thought leaders of our industry have gone unrecognized. Julie-Ann Williams correctly writes in this document that *"Mark Zuckerberg is recognized around the world but Barry Schrager and his compatriots walk unnoticed in most gatherings."*

I met Barry Schrager a few years ago when I became a volunteer of the SHARE Security and Compliance Project. I did not know at the time Barry started the Security Project in 1972 and was its first Project Manager. Shame on me. We as an industry are fortunate to have people like Schrager still involved in our industry, SHARE and its Security Project and should celebrate that fact. As you read this document, you will quickly understand why Schrager is in the Mainframe Hall of Fame (www.mainframezone.com/mainframe-hall-of-fame) along with other industry giants.

When we discussed the title of this document, I suggested using ***Mainframe Security – the Beginning***. Barry countered with ***The Beginning of Data Security*** because there was no security at all at that time. My mission is to make sure more people learn about the likes of Barry Schrager and his efforts to better secure data. I also feel it is important to help z/OS practitioners stay current while honoring those who paved the way for us.

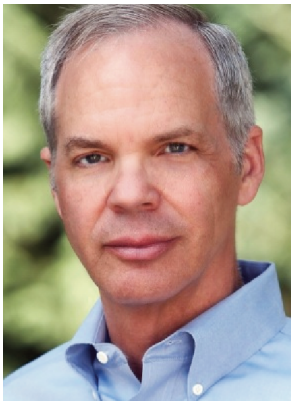
This is why I decided to use my company's resources to commission this document as well as previous books on auditing CICS (*CICS Essentials – Auditing CICS – A Beginner's Guide)* and *z/OS* (*z/Auditing Essentials – Volume 1– zEnterprise Hardware, An Introduction for Auditors*).

This document is the first article in *z/Auditing Essentials – Volume 2*, which will contain white papers from subject matter experts. We offer this first article ahead of the rest of the book because 2012 marks the 40th anniversary of the SHARE Security Project.

Jerry Seefeldt – December 2011

Introduction

It was 1974. The year that American racing driver Dale Earnhardt Jr, French rally driver Sebastien Loeb and British super model Kate Moss were all born but some of the "grown-ups" had other things on their minds...



Barry Schrager

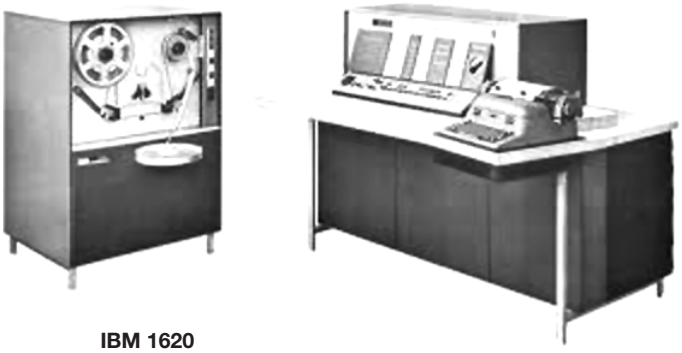
A group of IBM mainframe specialists gathered together to do one of the things they do best – think up new problems! It was time for Barry Schrager and his SHARE Team to report back on their research into this new problem they'd been looking into called data security.

It always sounds like a fabulous time to have been involved in this emerging technology "club" of mainframe computing. Educational establishments were leaders in that technology exploitation. Governments took an active interest in IT security and the Military (not to mention certain 3 lettered organizations) were involved in the same committees as us "normal folk". The stories that Schrager and others tell of the time put me in mind of the recent blockbuster movie, "The Social Network" except with more polyester and mainframes. Oh and all this was some 10 years before Mark Zuckerberg was even born.

Whilst Barry Schrager and some of his peers had an enormous impact on the way that we Mainframe Users do business today, their stories are not well known. Mark Zuckerberg is recognized around the world but Barry Schrager and his compatriots walk unnoticed in most gatherings. I hope to be able to do something to redress this imbalance!

This will be the first of a number of papers introducing a "Who's Who" of some of the brightest, under-sung stars of our mainframe generation. Following on from the acclaimed "z/Auditing Essentials VOLUME 1 - zEnterprise Hardware - An Introduction for Auditors" the papers are designed as a teaser for VOLUME 2 – Auditing z/OS. These folks will be providing their input to the new book.

I am blessed with being able to work with many of my heroes and you can look forward to a quick succession of interviews with the guys who weren't afraid to challenge the status quo. But, for now, come with me back to the 1960s for a bit of background on Barry Schrager's introduction to the cut and thrust of mainframe computing.



IBM 1620

After spending the Fall of 1964 at the University of Illinois at Navy Pier on the shores of Lake Michigan, the Campus moved to become the full 4 year University of Illinois at Chicago Circle. The new campus contained a Computer Center which supported the academic and research needs of the students and faculty.

Barry started his career working on the IBM 1620 (the scientific version of the 1401) and rapidly became the "Go to Guy" for many programming requirements – helping both his fellow students and the Faculty.

Barry's unique way of looking at IT problems came to light pretty quickly. As did his not insignificant skills in making a mainframe behave better. During one of our chats Barry shared the following story which I think demonstrates why he was able to be so influential.

> "I remember one case where a Faculty Chemist was working on molecular attraction and had a program that processed one piece of data every 14 HOURS! They used to start it on Friday evening and stop it on Monday morning. No one normally used the computer on weekends so this was not a problem, but everyone wanted a larger volume of data processed.
>
> So, I started working on making this task more efficient – first by doing approximations instead of calculating exact numbers from formulas, but that didn't help much. Then I just worked through the calculus and simplified the process, combining equations, etc.
>
> The outcome ran for just 45 minutes per set of data with the same results!
>
> I remember the Professor not really believing me so he sat me down and made me go through every equation, what I did, why, etc. After hours, he finally believed!!!"

This ability of Barry's to patiently explain things, sometimes over and over, until his audience understands, is something I have learned to lean heavily on. When even the most even tempered amongst us would be losing our collective cool, Barry maintains the same patience he started the process with!

He once found himself lent out to the Microbiology Department of Presbyterian St. Luke's Hospital which was close to the University campus for a project many of us will have benefitted from. It was a joint project with Argonne National Laboratories.

Argonne was working on a project "Automatic Chromosome Analysis By Computer" and their objective was to be able to develop a program that would pick out cells with damaged chromosomes from film shot through a microscope.

Argonne was interested in radiation damage to chromosomes and the Hospital in microbiological damage to chromosomes, but the process for detecting them would be the same. The geniuses behind this were a couple, Jim and Margaret Butler, researchers who worked at Argonne. The project was able to match chromosome pairs, and finding and pointing out any damaged ones, but, at that point in time they did not have a clue as to what the results meant. Barry takes up the story:

> "I worked on this and then left to go back to working at the Computer Center and never thought anything of it. Until about ten years ago when I first saw an Amniocentesis report...
>
> Amniocentesis is the test given to a pregnant mother-to-be which draws ammonic fluid from her and determines if the fetus has any genetic diseases.
>
> I couldn't believe it because the results I was looking at were the same as those we created at Argonne!"

Barry went back to work at the Computer Center. Partly because he was still a student but partly because they were getting "new toys"! An IBM/360 Model 50 for the geeks out there. He took a course in IBM 360 Assembler (taught by Jack Stoller – another of the unsung greats in this industry) and the die was cast. Despite his intent to go to Graduate School full time, Barry only stayed away for one semester. And it was then that our story starts to take shape.

One of the first tasks that Barry tackled on the new IBM/360 mainframe was the implementation of a product called Conversational Programming System (CPS). This was an interactive programming environment that compiled PL/1 into an intermediate code which was then interpreted when the student "ran" the program.

Fans of ACF2 may recognise the concept of a compile into a more efficiently interpreted state and then an interpreter which was inspired by this project.

Barry seems to have found it very hard to ignore a problem. As a result, he was a prolific developer of

special code to deal with difficult situations. A lot of this code still exists today!

For example the problem that the University had with a huge backlog of student jobs which were prioritized but never ran because higher priority jobs always came into the system. Barry's response was to develop the HASP Priority Ager which moved the jobs up in the queue at intervals. This code is still in JES2 today.

Barry got involved with SHARE at this time too. His initial interest was the TSO Project.

IBM/360 Model 50

In the early 1970's, Barry and Scott Krueger, who became one of the K's in SKK[1], needed something better than what TSO supplied for Job Submission and Output Retrieval. So, they created a product, which they describe as somewhat of a kludge, called the HASP/TSO Interface Package (H/TIP).

When it became time for the Computer Center to migrate from MVT to MVS, Barry, who was then Assistant Director of the Computer Center, had already assigned Scott to convert many of the HASP modifications to JES2. So Barry worked alone to write J/TIP (JES2/TSO Interface Package) which provided sophisticated job submission and output retrieval. It counted General Electric Timesharing, Mutual Life of Canada and even the IBM Development Labs amongst its Customers. The product is dead now but its spirit can be said to live on in SDSF.

Then there was this little problem MVS had with paging. It stemmed from there not being enough real memory to back up the virtual memory. One of the big contributors to this problem was a badly organized Link Pack Area. Barry had an idea:

> "I had an idea to collect a sample of execution addresses and optimize the placement of modules in the Link Pack Area. I asked Eb Klemens (the other K in SKK) to write some code to sample and collect the PSW addresses and then I used his data as input, with a map of the Link Pack Area, to determine where the high density execution was and, for modules less than 4K in length, pack the high usage ones on a small set of pages. The result was a packing list that significantly reduced paging due to the Link Pack Area."

During this whole time, security wasn't even a consideration. Computers were still, for the most part, independent devices not connected out over a wide area network. Access was primarily physical and no one had even thought about it. Sarbanes Oxley was a very long way off!

But, students started deleting and modifying Graduate Student and Faculty research data. So guess what happened? You got it – Super Barry to the rescue! Working with Eb Klemens again, the birth of the Hacker was soon being dealt with (albeit at a very superficial level on the mainframe to begin with). It was enough to see Barry being asked to form and become the Manager of the SHARE Security Project in 1972.

SHARE
Technology · Connections · Results

Key to the start of this project were both Bill Griffin from Mitre Corporation and Mary Lasky of the Johns Hopkins University Applied Physics Lab. They were both quite senior in the SHARE hierarchy.

The other members of the group, including representatives from Boeing and the Defense Department (the mists of time have dulled the memory and both Barry and I would like to apologize for being unable to either remember or find out who these strong hearted individuals were).

Barry presented back the results from the SHARE Security Project in a large SHARE Open Session in 1974. The representatives from the IBM Labs were in attendance. Bill Murray was the IBM representative to the project and he championed it within IBM.

When RACF was introduced in 1976, Barry and his Team at the SHARE Security Project initially thought that they had won a great victory. However, after closer examination, Barry realized that very little of the requirement, as defined in his 1974 paper, had been addressed. When he asked what had happened, he was informed that the IBM Development Team had decided that some of the requirements were not achievable! This included "protection by default" and "algorithmic grouping of users and resources". Think ACF2 pattern masking or RACF generic profiles to get the idea of the second requirement.

We know enough of Barry by now to realize that this kind of statement is simply a red rag to a bull! The result was that he went on to develop the Access Control Facility at the Computer Center which did satisfy the SHARE Security Project's requirements. After the University of Illinois declined to back the creation and funding of a commercial version of ACF, the London Life Insurance Company of London, Ontario, Canada funded it and ACF2 was born. It's a personal rule of mine never to say "No" to Barry!

Barry even tried to use this opportunity to complete his Doctoral Thesis. He had already picked Data Security as a topic because he was most interested in this new field. Barry picks up the story:

> "One of the members on my Committee told me that he wanted me to prove my theory in real life. That set me back because I felt there was no way I could prove this in "real life". And, IBM had just announced a $40 million program in data security so how was I going to compete with that?
>
> So, I worked on my concepts at the University of Illinois and turned them into the prototype for ACF2. London Life Insurance funded the development of the commercial version and then things moved very fast – General Motors Audit wanted ACF2 installed at one of their sites.
>
> This was in a world where, to take one example, GM's Delco Division had RACF installed for 18 months and had only 3% of their data protected. Their Auditors felt that they did not know exactly what percentage of data should be protected, but they knew that 3% was not it!
>
> In January 1978, the month before we went to London Ontario to develop the commercial version of ACF2, I gave a presentation at General Motors. The result of the presentation was interest and a statement by Corporate Audit – if you ever get this working, let us know.

So, after ACF2 was operating at London Life, GM, in an effort to put pressure on IBM to fix RACF, had Gerry Lyons, the Security Officer of Pontiac Motor Division come to London Life to see ACF2 and soon thereafter install it. (Sadly, Gerry Lyons has since lost his battle with cancer.) Pontiac was at 100% data protection in 3 months. With this early success at Pontiac, GM had their Assembly Division install ACF2. The Assembly Division was at 100% data protection in even less time.

Protected by ACF2...

Courtesy of GM Photo Store

GM Audit then had an awakening and wrote a letter to all their divisions demanding that they implement either

ACF2 or RACF. They went on to say, that because Delco Division had RACF installed for almost two years with only 4% of their data protected and Pontiac, GMAD, Chevrolet, etc had ACF2 installed for under six months and had 100% of their data protected, they wanted justification for how much data was actually being protected!

So, by the time I got back to the Dissertation Committee at Northwestern, a good percentage of the members had moved on. I met with the new Committee and told them that I did not have to prove my concepts worked in real life – General Motors and the Central Intelligence Agency had proved that for me. The new members felt that much of what I was saying was not really new, using patterns to select resources, and therefore not worthy of a dissertation!

Finally, they asked how many sites ACF2 was in and I responded 125. Their parting shot was to say that ACF2 was too commercialized to be a dissertation topic. They gave me six months to come up with a new topic. I told them I had too much going on to spend time with their foolishness."

Barry is an expert in both developing software himself and developing Teams to produce complex software. He is currently the Chief Security Architect at Vanguard Integrity Professionals, a supplier of RACF enhancement and other products to secure the z/OS Operating System. Over the years since our story takes place, he has started and run a number of software companies (including pivotal roles at SKK, EKC and JME) and has worked in many guises from Systems Software Developer through z/OS Security Analyst/Auditor and Team Leader through CEO. Whatever role he takes on, Barry brings his incredible level of technical understanding of the operating system and applies it in business terms. Under Barry's leadership, SKK developed the first z/VM security product, ACF2/VM (working with Charlie Kao) and the first MVS Operating System Auditing Product, Examine/MVS (working with Martin King). The later is now known as CA-Auditor.

But you can check out those facts for yourself if you are interested in finding out more about Barry Schrager. For me, he will always be my hero for never accepting "No" as an answer and for forcing an entire industry to listen to reason! He also did me the great honor of writing the foreword to "z/Auditing Essentials VOLUME 1 - zEnterprise Hardware - An Introduction for Auditors" and I am grateful to be able to return that favor!

I hope you enjoy Barry's original papers[2] which follow as much as I did. We are living in a world where much of the content is as fresh today as it was back in 1974. And I for one think that the world could only benefit from the wide-spread acceptance of lip prints as an authentication protocol, as described by Barry in his paper, – then I could start every working day by giving my lap top a quick kiss! ☺

My actual lip print – Julie-Ann

Note 1: SKK was the company that Barry Schrager went on to start (with Scott Krueger and Eb Klemens) in order to have a vehicle through which to sell ACF2 after the product concept went over so well at the 1977 SHARE Conference when he presented there. When SKK was sold to UCCEL (who were later acquired by CA), ACF2 was installed in about 2700 sites and has now generated over $1 Billion in revenue!

SKK led the development of mainframe security from the late 1970s. They also produced the first VM Security product, ACF2/VM and the first Operating System Auditing Product, Examine/MVS, now known as CA-Auditor.

Note 2: The original papers were produced using Wylbur on MVS!

Wylbur was a great text editing, program editing, job submission and retrieval system. In fact, the 1974 papers were written using Wylbur on an IBM 2741 terminal which was really a Selectric Typewriter connected to the mainframe!

I received the papers on actual paper and had to scan and OCR them. The electronic version has been manually compared against the original and I have tried to replicate the work exactly. Any errors probably stem from that process not mistakes by Barry Schrager!

SHARE VS/OS Security and Data Management Project

Goals for Data Security

The SHARE VS/OS Security and Data Management Project is holding this open session in order to acquaint the general SHARE membership with our goals for data security.

In San Diego, at the December 1972 interim SHARE meeting, this project started its investigation into the problems of data security. The group attending that meeting had very diverse representation, being composed of representatives from educational institutions, service bureaus, industry and Department of Defense installations. As the discussions progressed, we arrived at two conclusions:

- None of us were satisfied with the non-existent level of security provided by OS/MVT.
- We could not reconcile the different requirements of the group.

As I look back at it, the main problem we had was separating the issues of system integrity from data security.

Despite these divergent viewpoints we did agree on certain requirements of a security system. These are:

- The security system should be part of the operating system - not an add-on package
- Identification and validation of users is the first level of security
- The security-system should not be able to be turned on and off by an ”authorized user” - it should be continuously active
- The system should be able to run a highly secure job without having to purge itself of all other jobs or users
- The security system should be able to selectively invoke high overhead functions, such as dataset purging or rewriting with zeros, on an individual resource basis
- An interface program, for example IMS, should be supported as the only way to access a specific dataset

A short time later, IBM announced its forty million dollar program in data security. Then they announced VS2 Release 2, which provides the basis for a secure system – integrity. Suddenly the project was freed of the constraints of worrying about both system integrity and data security, and could focus its attention solely on data security.

Soon IBM will ship VS2 Release 2. For the first time a general purpose operating system that is guaranteed to have integrity will be available to the IBM user community. To quote the VS2 Release 2 Planning Guide's section on System Integrity:

> “A highly desirable property of an operating system is its ability to insure that one program cannot interfere with or modify another program’s (system or user) execution unless it is authorised to do so. For reliability and system availability, this is extremely important: for data security, it is essential. VS2 Release 2 provides this capability in the form of system integrity support.
>
> System integrity is defined as the ability of the system to protect itself against unauthorized user access to the extent that the security controls cannot be compromised. That is, there is no way for an unauthorized problem program using any system interface to:
>
> - bypass store or fetch protection, i.e. read or write from or to another user’s areas.
> - bypass password checking, i.e. access password protected data for which a password has not been supplied.

- obtain control in an authorized state.

  In VS2 Release 2 all known integrity exposures have been removed. IBM will accept as valid, any APAR that describes an unauthorized program's use of any system interface (defined or undefined) to bypass store or fetch protection, to bypass password checking, or to obtain control in an authorized state."

I assume that the severity level of these APAR's can range from severe to trivial, just like normal APAR's, depending on the severity for the installation involved. However, handling of the APAR's cannot be done normally. Placing the APAR in Early Warning Microfiche will jeopardize the security of every installation running the system. Yet to not notify the installations will leave then in a position of vulnerability. This is indeed a serious problem and we have not resolved it as yet.

Therefore, with the integrity commitment from IBM, it is now is an opportune time for us to develop our set of data security requirements for an IBM operating system.

Although physical security, protection from destruction of data through a natural or man-made disasters, disclosure of data via electronic eavesdropping, etc., are crucial issues in the total security picture, we feel that security systems from the operating system point of view are now an overriding point of concern. We are well on our way to solving the other issues, at least to the point where the risks associated with them are negligible compared to operating system security.

Pressure is also being brought to bear on us from other directions. More and more States, and the federal government as well, are enacting laws to prosecute those who disclose confidential personal data even if it is by negligence.

In addition, as we continue to consolidate and centralize our data processing centers and with the trend towards larger online databases, we increase our risks of accidental or malicious destruction and alteration of data. In many cases, we have dropped our manual backup systems. Even if we did have them, we have become highly dependent on computer based systems. For example, many companies have used the capabilities of computers to reduce their inventories to a minimum. Reverting to a manual system, even with accurate local inventories, could not be accomplished without a substantial period of chaos.

A few years ago, one could assume that someone might attack a database for personal gain. Unfortunately, nowadays, large corporations and governmental agencies may have their systems attacked solely for the purpose of disrupting service, with the attacker not really caring whether he is caught. Smaller companies are not immune either, since a revengeful employee could be similarly motivated.

We can no longer be complacent with the knowledge that we can probably apprehend all violators because even fear of apprehension is not always a deterrent. The motivation for the attack may be to interrupt service and have nothing to do with personal gain.

We must have adequate identification techniques and journaling systems that can allow us to reconstruct what happened. We must also have authorization and surveillance systems that will halt any attack at its early stages - before it can do any damage. Once the security system recognizes that an attack is in progress, it must take appropriate action to protect itself. Furthermore, the security system must be capable of tracking attacks which have failed and of initiating affirmative action to prevent further attacks from this source.

This action is obviously dependent on the installation and the data involved: but it may range from just notifying security personnel to actually shutting down the computer system itself. Whatever the case, the system must act reliably and predictably and must fail-soft, that is, if the system is to fail, it must fail in a predictable manner.

We believe that security must be a joint vendor-customer endeavour in that the operating system and all subsystems and application programs must work together in providing defense of data. Thus installations must develop security standards for all application software and provide mechanisms to enforce those standards.

Many times the argument is encountered that some other computer vendor has produced a security system

that is independent of the application programs. Where this is true, it is generally applicable only to single terminal/task relationships, introduces restrictions on sharing of data, and would require rebuilding our existing systems from the bottom-up. Multiple terminal/task systems, where the terminals have different security requirements, still require the cooperation of the application program to enforce security. Also, unfortunately, we have been developing our systems based on an operating system that was designed in the early 1960's and these designers just did not conceive of the applications and the associated security requirements of the systems that are in use today.

Most of us here have too much invested in application systems to be able to afford to start over and redesign our systems to run under a totally new operating system. Besides, many data dependent security constraints still could not be enforced without some security in the application program itself.

It is necessary to provide for auditability and administration of both applications and the system itself. Placing security controls in these areas, each independent of the other, could lead to obvious and severe inconsistencies. Consequently, in VS/OS Group Requirement #73-86, we have recommended the following:

Description:

There should be a centralized bank of resource control information and an installation replaceable operating system provided service for accessing and maintaining it. The resource control information must relate resources (such as datasets, program paths, etc.), conditions under which they can be made available (such as level of validation), and user identifiers. New types of resources, the resources themselves, the conditions, and the user identifiers must all be installation definable.

All authorization and delegation must flow through the single operating system access and maintenance service, and this service must be invocable during normal production operation. Invocation for the purpose of validating access to a resource should return a yes or no answer and optionally a variable length byte string to be used in corrective action (e.g. an error message, a module name, or a limit on a quantative resource). Security violation recovery routines should be modular and designed for easy user replacement.

Both Algorithmic grouping and grouping by itemization for both resources and user identifiers should he possible.

Incentive:

Demonstrable consistent application of resource control has become a requirement. In addition, administration of resource control will be facilitated by its centralization.

This facility, along with security standards in all "secure" application programs to make sure that they invoke it, solves the problems of consistency of application, ease of administration, and auditability.

Notice that in addition to physical resources such as datasets, the facility handles logical resources such as program paths and anything else the installation wishes to define. Thus an installation's security management will be able to say that an individual is able to enter a certain type of transaction from a set of designated terminals between certain hours. In this case, the transaction type can be considered the resource and the latter statements are the conditions under which the individual can access the resource.

The requirement also means that the information facility must be able to dynamically update its databank while the system is running. The system cannot be required to be unavailable during entry or processing of these changes.

Algorithmic grouping of resources and user identifiers is required; for example a group of datasets can be referred to as "SYS1.*". This means that, as datasets are added or removed from the system with the names beginning with "SYS1", they will automatically attain the protection level of the group without costly database updating for each addition and removal. This is particularly important in interactive environments where datasets are created and deleted continuously. The user and the installation must be able to specify global security requirements for these resources.

Along with this facility, an authorization system or application program is necessary to update the database. This program must be able to relate the system's secured resources to an administrator in the administrator's language. It must be easy to use by non-technicians, such as those responsible for defining

authorization privileges.

For “open shop” installations, it is necessary to introduce the concept of the “owner” of the resource where resources may be datasets and programs. The owner is the administrator of that resource and may specify who can access it. He may also request that accesses be journaled. This is particularly important when dealing with proprietary programs and databases.

All of the five requirements submitted to the project use the phrase “installation replaceable”. We define “installation replaceable” to include “easily modifiable in its source language”. Thus, the default routines provided by IBM should be designed for ease of modification, well documented, and written in a language for which we have a compiler.

A phrase that is often used in security systems is “acceptable level of risk”. We believe that using VS2 Release 2 as a base, the above requirement provides the authorization and delegation functions which are mandatory for a secure system and that installation security standards aided by the other project requirements submitted at the same tine can allow an installation to maintain whatever it considers to be its acceptable level of risk.

We feel that an installation is never totally secure and always has some chance of an attack against it succeeding. By devoting effort in the proper areas, an installation can reduce the probability of an attack succeeding, or can lower its level of risk.

We have determined five areas where effort must be placed to reduce the level of risk. Four additional requirements were submitted to IBM to give us the proper tools within the operating system to increase our security. One area is solely the installation’s responsibility. These areas are:

1. installation security standards
2. level of identification and validation of each user
3. journaling facility and recovery procedures
4. security violation recovery
5. designated interface programs

Installation security standards:

There is no way to underestimate the importance of adequate security standards and their strict enforcement. The programmers who code the application programs, the systems programmers who maintain and customize the operating system, the security staff that maintains and customizes the security violation and validation routines, all have an opportunity to subvert the security of the system.

As programmers and managers, we should welcome strictly enforced security standards and complete journaling or all operating system and application program changes. Systems programmers are in a particularly precarious position since they can be blamed for almost any incident on the system. Being able to affix responsibility for some action to an individual also means it is easy to exonerate everyone else.

However, we do not expect that this mechanism will get its greatest use from tracking down security violators, but rather that errors in the system will be more easily determined and corrected. Since all changes to application programs and to the operating system must be journaled in detail, they can easily be backed off. Improved maintenance procedures are a significant by-product of security standards.

The minutes of the project meeting last August reflect this concern:

> “An important point was raised about the system database (e.g. LINKLIB, LPALIB, etc.). This is one of the most critical databases on any system and a full audit trail is absolutely necessary both in case of eventual security violations and standard recovery in case of system failure.”

IBM, in a recent announcement, has given us a tool called the System Modification Program which is due to be shipped at the end of this month. This service aid applies superzaps, module replacements, macro replacements, and source code changes via IEBUPDTE. It keeps a detailed log of all changes and provides a back-off capability for removing modifications.

Level of identification and validation:

Another area of risk in a secure system is “How sure is the system that a user is who he says he is?” This problem is compounded by the use of different identification and validation procedures in batch, TSO, APL, IMS, etc. Group requirement #73-85 aids an installation in reducing its risk in this area:

Description:

All subsystems must call upon a single installation replaceable subroutine for the purpose of identification and validation of all users of the system. A default routine should be provided by IBM for this purpose along with full specifications of the interface requirements.

The subsystem should provide to the subroutine the origin of the entry request and full facilities for interacting with the user and related equipment.

Incentive:

Identification and validation of the user is a prerequisite to any data access control system.

Comment:

The purpose of this requirement is not to ensure identical logon protocols but rather to specify that the information collected and the validation procedures are identical for both batch and interactive systems.

This requirement gives the installation the ability to validate the identity of the user to whatever level it wishes and to use this level of validation in later security access decisions. For instance, with hardware now readily available, it is possible to have the following levels of validation:

- Password only
- Password/Badge reader
- Password/ 2 Badge readers - one being used by a guard who has visually identified the user.

More exotic validation techniques could use fingerprints, voice prints, lip prints, etc. Identification and validation of a user's identity is the cornerstone of security. This, combined with proper journaling, provides the ability to limit the accountability for some action to an individual. It is important to think of the initial identification and validation procedures as being the first line of defense of the system. The risk of data security violations is reduced, by limiting access to the system to only those who are authorized to use It.

Journaling facility and recovery procedures:

Application designers must be encouraged to journal everything that is necessary to reconstruct an event or recover a database. Journaling should be a service offered by the operating system just like the access methods. Depending on the installation, the journaling facility may have to handle very large volumes of information and utilize several output devices. It may have to have the strictest integrity and use special hardware such as a special tape drive and controller that buffers the output within the controller so no data can be lost by a system failure. Another possible hardware requirement is a “write once” tape that cannot be rewritten.

The main requirements of a journaling facility therefore are:

- easy to use
- able to handle large volumes of data efficiently
- be tailorable to an installation's needs
- provide an unimpeachable record of activity for auditability and recovery

VS/OS Group Requirement #73-87 requests this facility:

Description:

There should be a centralized installation replaceable journaling facility provided by the system control program which can be invoked by any program. This facility must be capable of recording on a variety of data storage devices. Where IBM provided subsystems use this facility, programs should be provided by IBM to analyze these entries.

Incentive:

A centralized journaling facility is necessary for auditing and recovery.

Another area of concern is a misuse of authorization. For example, a person may be authorized to access certain types of services, but by combining them in an unforeseen manner, he would be able to do something that is not authorized.

Detailed journaling by the application programs and by certain operating system interfaces would allow reconstruction of the sequence of events which led to the unauthorized use.

Security Violation Recovery:

It is admitted by most security people that, given enough time, any security system can be violated. However, in the process of testing a system's defenses and attempting to bypass them, indications of these attempts will occur. Whether the system will actually be violated, is therefore a function of how alert an installation's security staff is, how good the installation's security violation analysis programs are, how well attuned the security system is to the installation's needs, and what action the system takes to preserve its security.

A security violation facility should be provided by the operating system in order to institute timely and appropriate recovery action. Group requirement #73-88 addresses this need:

Description:

The system control program should provide an installation replaceable security violation exit. An IBM default routine should be provided which makes a journal entry and writes a message with a security violation routing code.

Incentive:

Timely corrective action and notification of the proper installation personnel is critical. This is an extremely installation dependent function.

A trivial example of this requirement is the case of several failures on a prompt for a password at logon. Most systems merely disconnect the line - a secure system should call upon the security violation exit which will deactivate the user-id until security personnel have investigated.

Designated interface programs:

The project's final requirement deals with designated interface programs. We do not feel that it is the responsibility of the operating system to do "record and field" level data protection although we do feel that the operating system should maintain the authorization database which is involved in making those decisions.

The possibility of an application program inadvertently modifying data incorrectly and the chance that an authorized user of a database may bypass security and journaling functions by using his own program to access a database led us to group requirement #73-89:

Description:

There should be the capability of associating with any dataset a single interface program capable of accessing that dataset. Where the interface program is a subsystem (e.g. IMS) an interface should be provided to other subsystems (e.g. TSO).

Incentive:

The need to be able to limit the path to a dataset to one interface program structures the system so as to provide increased integrity, security, and back-up capabilities.

This requirement also addresses the problem of data integrity. Data integrity is concerned with the accuracy and completeness of the data while data security is concerned with protection from unauthorized destruction, modification, or disclosure whether accidental or intentional. Misuse of authorized privileges, errors in application programs, etc. can cause a database to lose its integrity. In these cases data security, as we define it, has not been violated.

A designated interface program is in a position to perform validation of database update requests and to journal all changes. This helps to protect the integrity of the database by providing a record of database modifications for error determination and recovery. In addition, it can be much more efficient than the operating system in journaling those chances since it is able to deal with specific and meaningful information.

The operating system would have to journal all data changes, thereby incurring much more overhead.

In summary, VS2 Release 2 provides a base for a data security system because of its integrity features. The VS/OS Security and Data Management Project has determined the following requirements for a data security system:

- Security system should be an integral part of the operation system and should be tailorable to an installation's needs.
- High overhead security functions should be selectively invoked on an individual resource basis.
- Users should be identified and validated in the same manner at all interfaces to the system.
- Resource control information and decision making should be centralized.
- Journaling capability should be an operating system service.
- Security violation handling should be centralized.
- Designated interface programs should be supported on a dataset basis.

The five requirements submitted by the project on this issue were:

- 73-85 Identification and Validation Exit
- 73-86 Central Resource Control Information Facility
- 73-87 Centralized Journaling Facility
- 73-88 Centralized Security Violation Exit
- 73-89 Designated Interface programs.

If anyone has any questions, opinions, or suggestions concerning these topics or any other aspect of data security, please contact me or any or the other project members.

Barry Schrager UIC
Project Manager
March 4, 1974

IBM Data Security Forum

Centralized Resource Control Information Facility
Barry Schrager
Assistant Director
Computer Center
University of Illinois at Chicago Circle
Chicago, Illinois

IBM Data Security Forum
Denver, Colorado
September 10-12, 1974

In December 1973 the SHARE Security and Data Management Project submitted requirements to IBM in the area of future data security goals. The objective of these requirements was to define an environment within the OS/VS operating system such that a basic security system would exist and implementing a customized security system would be a feasible task for most installations.

IBM has provided a crucial prerequisite of this environment with OS/VS2 Release 2 and its system integrity support. Until this time, the only way to provide a secure system was to limit system access to some secure subsystem. An example of this type of subsystem could be IMS. Unfortunately, the security of such a system leaves much to be desired since most installations, because of economics, are forced to allow application program development and system programming activity to simultaneously occur on the same system as the hopefully secure DB/DC subsystems. These concurrent activities, all occurring on the same data processing system are an area of considerable concern.

This is not meant to indicate mistrust of the system programmers; rather, this concern is a realization that when there is a breach of system security, those with unrestricted access to the system are, unfortunately, often assumed guilty until proven innocent. We define the datasets which contain the modules of the operating system as an entity called 'the system database' and system programmers must deal with this database daily. This database is the most important database on any system, since a violation of its integrity can lead to an undetectable violation of any other database on the system. To treat the system database lightly, or with inadequate, security and journaling procedures, is a clear invitation to disaster.

It is my belief that the first objective of a security system must be to protect both the system database and the application program database from unauthorized change. This includes the automatic journaling of sufficient information such that recovery of the system to the state prior to the change is possible, as well as making a record of each change so that accountability rests with a single individual. Furthermore, the journal itself must be completely secure and inviolate. Good maintenance procedures are a by-product of good security procedures since even unintentional errors can be easily found and corrected.

The ability to identify the responsible party for any modification to any system, application, or working database on the data processing system is a requirement. To be able to do this implies that the system must be able to identify its users with a reasonable degree of confidence. The degree of confidence should be related to the importance of the database and the level of risk that the installation wishes to assume. This requirement is currently complicated by the fact that today's delivery subsystems such as JES, CICS, IMS and TSO, have no common method of identification and validation of that identification. In fact, some do not provide for any identification at all.

It is for the above reasons that one of the SHARE Security and Data Management Project's requirements called for a common user identifier across all interfaces to the system, and for the ability of the installation to provide its own validation techniques in order to assure the desired degree of confidence in the identification necessary for their environment.

An installation's validation techniques may include the use of passwords, badge readers, or the placement of terminals and other input devices in supervised areas where a secondary visual identification is required by a guard before access is permitted. The required hardware for this support is currently available and the software required is not difficult to produce; yet almost no delivery systems support it. The point is that we have no system-wide user identifier and no software support of the hardware security devices.

It should also be noted that an unsuccessful attempt to access the system may be an indication that the system is under attack. Unsuccessful attempts should be journaled with all available supporting information, including the geographical location of the attempt if it is available. Computer analysis of the journal may provide an installation with information that can help to prevent successful unauthorized access to the system.

Likewise, since the user identifier is the same for all delivery systems and is unique to a single individual, simultaneous access to the system with the same identifier from two terminals is an absolute signal that the security of the system has been breached and security personnel should be immediately dispatched to investigate. Simple computer analysis of all access attempts may also uncover violations of the system security if the accesses from a single user occur from different geographical locations, and these accesses are physically impossible to accomplish within a short period of time.

Thus, common user identification, sufficient validation techniques, and some method of restricting users to only those subsystems that they need to access are the first lines of defense for any system. This, along with proper journaling techniques, which can limit the responsibility for any action to the level of an individual, which, if well publicized, can act as a deterrent to misuse of authorized privileges, and which can reconstruct the system to the state prior to an undesirable modification, may be all that an installation requires to protect its mission capability.

These procedures are sufficient protection for an installation only under limited circumstances. First, data processing management, either directly or indirectly via corporate management, must have absolute control over all users of the system, such that it would be extremely disadvantageous for someone to misuse his authorized privileges. The second condition is that any unauthorized alteration of data could be quickly found and corrected, and responsibility for this alteration could be easily assigned. The third condition is that the mission capability of the system would not be seriously jeopardized during the interval of time between the alteration and the eventual correction of the data.

The procedures discussed are not adequate if there is no direct control over the users such as in a service bureau environment, if the confidentiality of the data is such that serious consequences would occur if disclosed, or if the correctness of the data at all times is essential to the mission capability of the data processing center.

An additional step towards protecting data would be a program such as IMS which would be supported by the operating system as a single designated interface for accessing a database. This implies that the "designated interface program" must be able to analyze the access request to determine its validity both in terms of the individual making the request and the content of the request. An interface program should also be capable of preventing accidental destruction of the data and be in a good position to efficiently journal changes to the database for later recovery.

However, although designated interface programs are required for system security, to place the authorization decision making within each interface program may be difficult and lead to inconsistencies between various programs. In addition, as was stated previously, if this is to be considered a solution, then the system and application program databases must also be treated securely. The only program and process that is able to protect these databases is the operating system itself, and no preconceived security scheme provided by the vendor will be able to enforce the wide variety of requirements found in various installations.

A good example of this is the question of whether to have centralized or decentralized administration of the

security or the data on the system. If the system only services one corporate function, if the corporation believes it is economically and practically feasible to have a central security staff, or if the system handles highly classified information, then a centralized staff for administration of the security function provided by the data processing system is mandatory.

In a service bureau environment, a university, or anywhere that the responsibility for the security and integrity of the data lies outside of the data processing center, then the people who own or are responsible for the data must have administrative control over it. In these kinds of environments decentralized administrative control places security where it is most meaningful.

Any security system must be able to function regardless of whether its administration is centralized or decentralized. It must be a part of both the interface programs and of the operating system if it is to be effective.

In addition, the security system must be fairly efficient. This is not to say that it must induce no overhead, but rather that the overhead induced must not be so great that it is not economically feasible to use the system. Overhead can be reduced by decreasing the number of discrete security decisions that the system must take. It is better to control more global resources such as transactions, rather than to control local resources such as the fields within a database.

The case against controlling local resources can also be strengthened by what can be loosely described as a cross-coupling of resources. An example of resource coupling would be that, while it may be permissible for a user to modify a field in a database, a condition imposed may be that some other field is also modified in a predetermined manner. This would require a sophisticated designated interface program or simple transaction level authorization. In this case, control at the transaction level would reduce overhead, simplify the analysis of the system and not jeopardize the security of the data.

Unfortunately, as the magnitude of the resources being controlled increases, the proportion of responsibility for the implementation of the security system shifts from the vendor to the customer. For example, while the responsibility for the limitations of access to specific transactions may lie with the vendor, what the transaction does and what rules it follows are the customer's responsibility.

If the data and programs residing in a data processing system are considered the resources of that system, then the function of the security system is to control the interaction between the users of the system and these resources in a manner predetermined by the administrators of the resources.

Whatever the implementation of this system, it must be easy to use and easy to understand. For many systems, a high requirement is that it be easy to audit. Complexity and confusion within the system may make the auditing process difficult and therefore lead to unknown irregularities.

However, an uncoordinated approach, with a multiplicity of resource control routines in the various interface programs, in the operating system, and in other miscellaneous application programs, is neither easy to use nor easy to understand. Understanding the complex interaction for several systems, each with its own unique database and algorithmic processes, will make comprehension of the system as a whole difficult at best. Finally, demonstrating consistent application of resource control will be time consuming and difficult.

On the other hand, a single process within the system, making resource control decisions, must treat resources consistently and be easily extendable for new applications. Its interfaces to the system should be modifiable so that, with simulation, its decision making processes could be more easily tested, understood and verified. With a well-planned set of interfaces via the system control program, it would be easy to use for application programs. Since it would be removed from the application programs themselves, application programmers need not know the exact decision making process that would be used. Conversely the decision process could be easily modified without having to modify each of the application programs. And finally, since it would be removed from the physical resource control, it could easily control conceptual resources such as program paths.

Examples of program paths are transactions, command sequences, and operating system processes such as "open". A program path can also be defined to include the flow of control within a module. This enables an installation to define different security levels for different paths within an application program, without having to rewrite different application programs due to the differing requirements for security. It also avoids

the need for coding authorization checks within the application program itself.

All authorization decisions would be made within a central control facility which would be a service function of the operating system. The central facility for resource control would make decisions when requested by other parts of the operating system or application programs. Implementation of the decision would be left to the requestor. Possible responses by the central facility could include a yes or no answer, a limit on a quantitative resource, an error message to be displayed to the user, or the name of an error routine to which control should be transferred.

The title of this paper, Centralized Resource Control Information Facility, is the title of a SHARE Security and Data Management Project requirement which states:

> "There should be a centralized bank of resource control information and an installation replaceable operating system provided service for accessing and maintaining it. The resource control information must relate resources (such as datasets, program paths, etc.), conditions under which they can be made available (such as level of validation), and user identifiers. New types of resources, the resources themselves, the conditions, and the user identifiers must all be installation definable.
>
> All authorization and delegation must flow through the single operating system access and maintenance service, and this service must be invocable during normal production operation. Invocation for the purpose of validating access to a resource should return a yes or no answer and optionally a variable length byte string to be used in corrective action (e.g. an error message, a module name, or a limit on a quantitative resource). Security violation recovery routines should be modular and designed for easy user replacement.
>
> Both algorithmic grouping and grouping by itemization for both resources and user identifiers should be possible."

The incentive given for this requirement was that "demonstrable consistent application of resource control has become a requirement." In addition it stated that administration of resource control will be facilitated by its centralization within the data processing system.

Having a central facility, such as that described above, means that more effort can be spent in making it a sophisticated product. And sophisticated it must be, if it is to be useful. Algorithmic grouping of resources is an example of this sophistication. In addition to itemizing resources, the requirement states that resources must be able to fit into groups as described by some algorithm. The TSO notation of asterisk for dataset naming conventions to indicate all qualifiers as in Userid.PROGRAM.* is an example of this. This means that the security information database need not be updated each time a dataset is created or deleted and will automatically assure whatever level of protection that is specified by the algorithmic group that describes it.

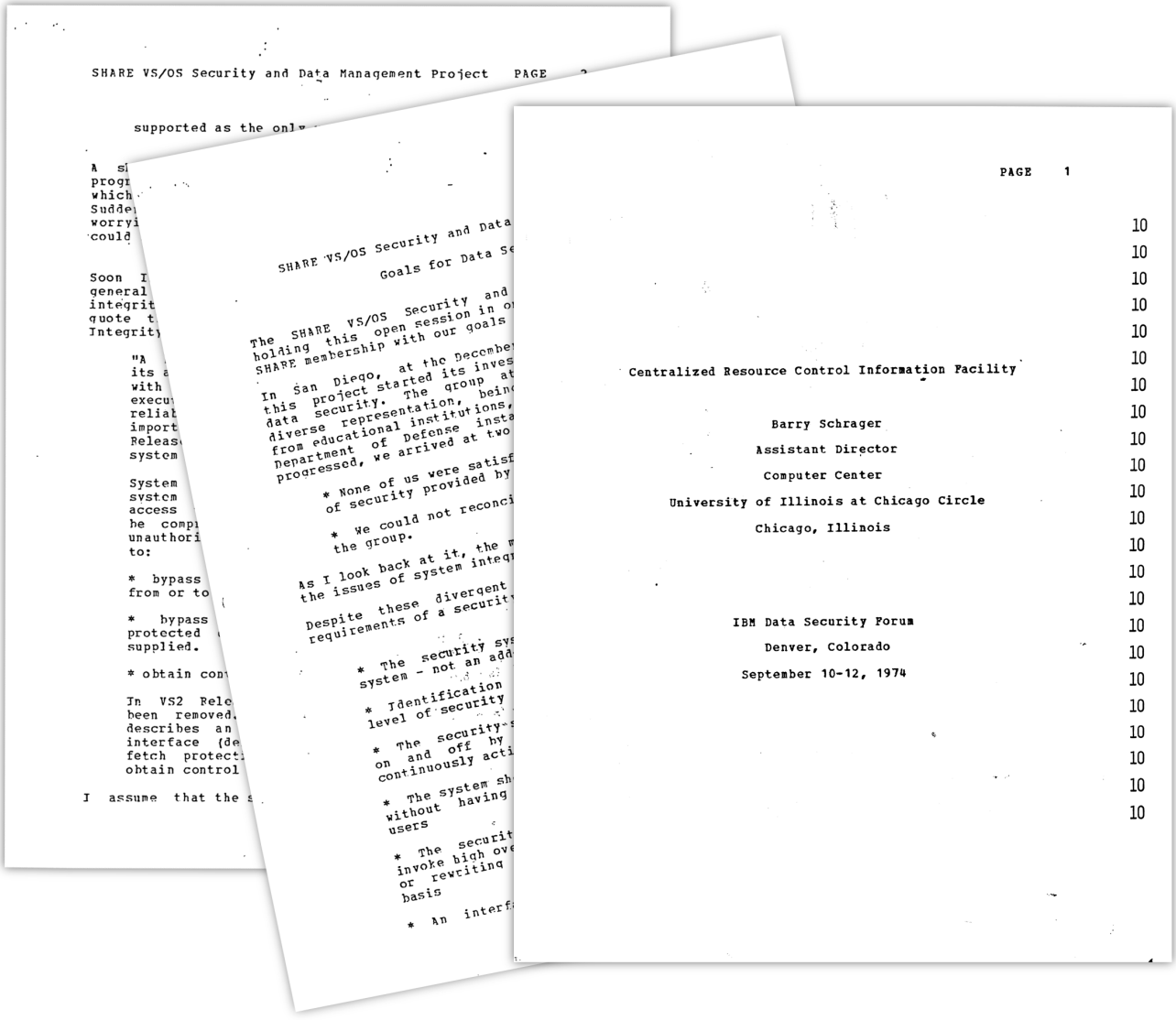
The implications of the system described are interesting. Security is a joint customer-vendor endeavor. It is the responsibility of the vendor to supply the services within the operating system to access the centralized resource control information facility; to provide an easily extendable set of routines that comprise the facility for the basic decision making of the operating system; and finally to provide the capability for performing maintenance on the information used by the security system. With vendor supplied delivery systems, such as IMS or CICS, security support should be provided as additions to the central security facility rather than as independent routines. And finally, support for a system-wide user identifier and an easy-to-use journaling facility are required

It is the installation's responsibility, however, to verify that their own application programs formulate requests correctly to the facility and take the correct action as specified in the facility's response. This means that application programs should formulate their requests to the central security facility in a manner consistent with the operating system's use of this facility. This is an obvious example of the consistency of application criterion for security rules.

Although it is not entirely satisfying to have a security system that is neither totally the vendor's responsibility nor the customer's, this set of recommendations seems to offer the best base for continued extendability. We must acknowledge that this is an era when systems are constantly being used for new applications at

the same time that there is constant pressure for consolidation of facilities and databases. These pressures are in turn forcing less specialized and more general purpose operations.

The concept of the centralized resource control information facility, along with a system-wide user identifier and proper journaling techniques, is capable of simultaneously satisfying the requirements of a batch system, an interactive timesharing system, and a data base/data communications system. It is time to develop a coordinated system-wide approach to the solution of our resource control problem.



SHARE VS/OS Security and Data Management Project PAGE

supported as the onl

A si
progr
which
Sudde
worryi
could

Soon I
general
integrit
quote t
Integrity

"A
its a
with
execu
reliab
import
Releas
system

System
system
access
he compl
unauthori
to:

* bypass
from or to

* bypass
protected
supplied.

* obtain con

In VS2 Relo
been removed,
describes an
interface (de
fetch protecti
obtain control

I assume that the s

SHARE VS/OS Security and Data

Goals for Data Se

The SHARE VS/OS Security and
holding this open session in o
SHARE membership with our goals

In San Diego, at the Decembe
this project started its inves
data security. The group at
diverse representation, being
from educational institutions,
Department of Defense insta
progressed, we arrived at two

* None of us were satisf
of security provided by

* We could not reconci
the group.

As I look back at it, the w
the issues of system integ

Despite these divergent
requirements of a securit

* The security sys
system - not an add

* Identification
level of security

* The security-s
on and off by
continuously acti

* The system sh
without having
users

* The securit
invoke high ove
or rewriting
basis

* An interf

PAGE 1

10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10

Centralized Resource Control Information Facility

Barry Schrager

Assistant Director

Computer Center

University of Illinois at Chicago Circle

Chicago, Illinois

IBM Data Security Forum

Denver, Colorado

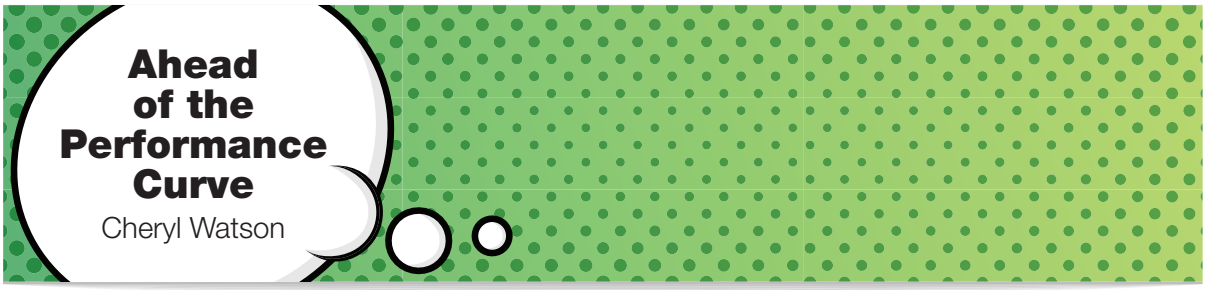
September 10-12, 1974

Continuing the Work

The SHARE Security and Compliance Project continues to advance the practice of better data security. Project volunteers as on February 2017 include:

Carla Flores – CA Technologies – Project Manager
Julie Bergh – IBM Corporation
Marlaina Chirdon – Vanguard Integrity professionals
Ross Cooper – IBM Corporation
Phil Emrich – Vanguard Integrity Professionals
Robert Hansel – RSH Consulting
Brian Marshall – Vanguard Integrity Professionals
Charles Mills – Correlog, Inc.
Phil Noplos – American Express
Paul Robichaux – NewEra Software, Inc.
Roxane Rosberg – Vanguard Integrity Professionals
Jerry Seefeldt – NewEra Software, Inc.
Phil Smith – HPE Data Security

The SEC Project encourages all security professionals to join the Project. More information is available on the SHARE Security and Compliance Project website – www.SHARE-SEC.com



Foreword

Although it has been my real pleasure to know Cheryl Watson for more than 23 years, I didn't really know much about her. For that reason I was thrilled to learn that she was being profiled in the new book, *"Cheryl Watson – Ahead of the Performance Curve"*. It's in this book where I really discovered that she is not only the sweet lady I had the privilege to see just occasionally at CMG or SHARE conferences, but that she also was a tireless worker who was always eager to learn about new technologies that came her way. Hopefully, with the exposure of this book many others will come to appreciate what a huge difference Cheryl Watson has made in improving the performance of the IBM mainframe for so many years.

It was shortly after I started publishing a magazine named Mainframe Journal that I met Cheryl while we were both attending a CMG (Computer Measurement Group) conference. I regularly attended CMG because it was the perfect conference to promote *Mainframe Journal* as it was almost entirely focused on the performance management and capacity planning issues facing users of IBM mainframe computer systems. The attendees were prime candidates to become subscribers and the speakers were all ideal candidates to write the technical articles I wanted to publish. And Cheryl Watson was at the top of my list! So I was thrilled when she gave me an article to publish titled, "Why Isn't a CPU Second Consistent?" That article turned out to be one of the most popular articles we ever published. From that point on, I felt honored any time we were able to feature a Cheryl Watson article.

Over the succeeding years I have witnessed the rise of Cheryl Watson to virtual mainframe rock-star status. But to me she is much more; she is a very sweet lady I am proud to call "friend". She also helped to elevate *Mainframe Journal's* reputation when she was kind enough to allow me to publish her exceptional articles.

In August 2010, I had the privilege to add Cheryl Watson to the Mainframe Hall of Fame because of the way she has shared her extensive knowledge of performance analysis and capacity planning since 1965 as a consultant, teacher, and author of "Cheryl Watson's Tuning Letter."

Bob Thomas
President, Enterprise Systems Media, Inc.
Publisher of Enterprise Management & Enterprise Tech Journal
Dallas, Texas

Performance Management

This is the second paper introducing a "Who's Who" of some of the brightest, and yet under-sung stars of our mainframe generation. These folks have kindly agreed to be interviewed and will be providing their input to the new book, "z/Auditing Essentials VOLUME 2 – Auditing z/OS" that follows on from the acclaimed "z/Auditing Essentials VOLUME 1 - zEnterprise Hardware - An Introduction for Auditors".

I waxed lyrical about being able to work with many of my heroes in the last paper and this time is no different. Cheryl Watson is unique in the mainframe industry. There are folks out there who have heard of "Cheryl Watson" but know it only as a corporate identity not as a person! This woman stands out from the crowd, not just because she is a woman (unusual enough on its own in this business nowadays) but also because she has helped to shape the systems that we work on since early in her career. Cheryl was one of the first people who challenged IBM's newly issued performance figures for CMOS (**C**omplementary **M**etal **O**xide **S**emiconductor) based mainframes when they were first released. She continues to blaze an influential path today…

Cheryl Watson-Walker

You may be wondering what an interview with the woman who is most famous for questioning IBM's mainframe performance figures is doing in a book about security...

Wikipedia say "Big Data is a term applied to data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time."

And Big Data is exactly what we are securing and auditing on System z. Even restrained estimates still suggest that 70% of the world's business data sits behind a mainframe. The world has at last caught up and now the true value of data is being realised.

It is equally important to make sure that our Big Data can be reached in a timely manner as it is to make sure that it is secured against misuse.

As Barry Schrager was introducing the mainframe world to the idea of Data Security, Cheryl was making the dark arts associated with tuning the performance of these beasts accessible to those of us with enough understanding to have been only dangerous before!

I first came across the Cheryl Watson Tuning Letters as a nervous young MVS Systems Programmer! But more of that later, I want to start at the beginning so come with me to 1965 the year our heroine first gets involved in mainframes.

Born in Portland, Oregon (never ask a Lady how old she is!), Cheryl was something of a Nerd at High School. She'd always liked Math but accessible computers were still some way off, so the term Geek really hadn't taken root yet!

She majored in Math at Portland State University (with a minor in Physics) and married whilst still in college. So when her husband decided to get his PhD, Cheryl had to abandon plans to get her Masters in favor of getting a job to support the two of them. Thanks to her selfless devotion to family, the mainframe world gained one of its brightest stars, although no one would have suspected it at the time!

Cheryl joined Consolidated Freightways as an Applications Programmer in 1965. They had both an IBM 1401 and the mainframe version, the 7010.

The latter had a massive four I/O Channels and 100,000 characters of memory in which to operate all instructions. It cost $1,307,550 back in 1964!

She initially learnt to write Autocoder but, when Consolidated moved to S360 in 1966, changed to COBOL and eventually moved into the Systems Programming team.

Cheryl had found her niche! In fact, she stayed in this team for three years and only moved on when her husband found a new job and she went with him.

Interestingly, a pattern started to emerge. Cheryl found a job working with one of her new neighbors who was in the process of setting up a software company to produce a 4GL compiler. This wouldn't be the last time that Cheryl made career progress not because of a resumé (or CV) but because of the personal contacts she has made. In fact, even today, Cheryl has never had a regular resumé! It's a testament to her determination to succeed that she has never been asked to conform to the normal expectations set upon employees.

Exploiting her tendency to make the best of any bad situation, Cheryl turned to her advantage the simultaneous collapse of her marriage **and** the company she was working for, after a chat with an old friend. This former colleague was working for ISM (Information Systems Management) when he convinced Cheryl to move to Phoenix, Arizona to take up what would turn out to be the first of many "Dream Jobs" as a traveling Trainer for the software that ISM supported.

If there's a better way to get over a messy divorce than traveling the world being treated as an expert on many different things, I have yet to find it! So Cheryl traveled and had fun while she worked. She was teaching out in Europe (Holland to be precise) when ISM (there is now another organization using the same name) sadly went out of business!

If Cheryl wasn't such a lovely example of a human being it would be impossible not to dislike her at this point…

While she was sitting in a café (No not one of those café's) in Holland, wondering how she was going to get home, ISM was bought by an organization based out of San Francisco. They had heard of Cheryl and what she was doing for ISM and they REALLY wanted her on board. They arranged flights back to the US for her to go straight to San Francisco, packing up and moving her entire life to be there when she arrived. Now **that** Ladies and Gentlemen is some welcome!!

Quite rightly, Cheryl LOVED being in San Francisco and was there for a little over a year before she was head hunted by EDS (Electronic Data Systems founded by H. Ross Perot). The wonderful thing about having had a varied career is that you have interesting answers when people ask you questions. So a formal chat with her peers turned into yet another high profile, career enhancing job offer!

She worked for EDS (on and off) for over 10 years and had some very interesting engagements during that time. I've picked just a few examples here.

In 1971 Cheryl was placed onto the EDS team that was developing a new Claims System for Blue Cross/Blue Shield. She started on the project as a COBOL programmer but immediately demonstrated her uncanny ability to poke around under the covers of the system she was working on and unsurprisingly was swiftly moved into Systems Programming!

Whilst there, Cheryl took another one of those odd calls that she gets... Amdahl wanted her to go and work with them in California so she went. Despite only being there for 6 months she says: "I learnt **loads!**" But she soon returned to EDS in San Francisco. It was while she was there that she met and married her second husband.

They were then moved to Dallas by EDS for Cheryl to work as a Trainer in their Systems Engineering Development Scheme. This was a three month long, total immersion training plan and started on day one with machine code and Assembler. Some people questioned this sequence but Cheryl defends her position by saying: "If we started with Assembler on day one then by the time we got to COBOL, the students could already code and more importantly, read dumps!"

Cheryl realized quite quickly that her real skills lay in a pure technical direction. Whilst she taught Management Skills, she freely admits that she never learnt the lessons for herself! However, EDS always allowed promotion through a technical career, not just by moving into management, so it was never a problem.

Cheryl and her husband returned to San Francisco to help to get a troubled project back on track. Whilst there, EDS wanted Cheryl to help with the recruitment of a full team of mainframe professionals to work for Allstate Insurance in Northbrook, Illinois. They asked her to interview candidates. She had been out to visit with the Customer three or four times when she realized that one of the jobs she was recruiting for was one that she would just love to have for herself! It was the CICS Administrator/Systems Programmer role, which involved teaching and many other things that Cheryl enjoyed. So she quite blatantly poached the job, moved to Illinois and had two years of having a surprisingly fabulous time despite her second divorce. She retook her maiden-name of Watson and prepared to launch "Cheryl Watson" on an unsuspecting public!

Cheryl stayed until the end of the project and got her first taste of Performance Management after which EDS moved her back to Dallas in 1978.

RMF (Resource Measurement Facility) was in its infancy but when Cheryl saw its predecessor, MF1 (Measurement Facility 1) she just thought it was the neatest thing she'd ever seen! She eventually did a deal to exchange a large amount of unused vacation time for attendance at her first CMG Conference (Computer Measurement Group). User Groups can have a dramatic impact on people when they first discover them and Cheryl was not immune. She reports being "completely blown away" after hearing Barry Merrill talking about one of the first Performance Monitoring tools, MXG (Merrill's Expanded Guide which was, and still is, based on SAS).

Cheryl wrote up a 100 page report on her attendance at CMG which she shared with all of her colleagues. She started with a cover page summary, followed by 2-3 pages titled "Things We Should Be Doing". Then each department had its own section, which incorporated both a summary of the relevant presentations and a copy of the handouts provided at the conference. She also gave classes based on information she had learned at the conference!

It might seem like a lot of effort to go to but these actions guaranteed that Cheryl's name was top of the list when her employers were looking to send someone to a conference after that. She started volunteering with various User Groups in the late 1970s.

Cheryl takes up the story a while later:

> "One of the best times I had working came in 1980s when EDS sent me to the Netherlands to work on an outsourcing assignment. I was able to do all of the things I loved, such as teaching, performance, and capacity planning, as well as attend some of the European conferences.
>
> That's where I re-connected with John McCann, who I had worked with at SHARE in the CME (Computer Measurement and Evaluation) project. That was a happy meeting, because a year later, John hired me to help start up the German office of Morino Associates (founded by Mario Morino). That was almost two years of living half-time in the UK and half-time in Germany, and traveling throughout Europe teaching the Morino products and providing technical sales support.
>
> It was a dream job, and let me see so much of Europe and develop some great friendships. That's also where I learned SAS for the first time."

Cheryl returned to the USA when Morino (later merged with another company to form Legent Corporation in 1989 and subsequently taken over by CA in 1995) wanted her for their Education Department. And so she started teaching products like TSO/MON (a TSO performance manager product) and MICS (MVS Integrated Control System - a SAS database product now marketed by CA). Cheryl is fiercely loyal and I love the way she always suffixes TSO/MON with the statement "Still one of the greatest products I've ever seen!"

This high profile job saw Cheryl meeting with hundreds of people in the Performance Management and Capacity Planning fields. But after a while the call of the deeply technical became unavoidable and she moved into development of a major release of MICS. It seems that Mario Morino was a very influential person in Cheryl's life. She says: "His passion for the field was infectious!"

So infectious it seems that Cheryl was soon working 50-60 hour weeks at Morino Associates Inc!

It was during this particularly intense period that Cheryl met the man who turned out to truly be her soul-mate, Tom Walker. Cheryl is unapologetic about championing her relationship with Tom and she has good reason for that. But I'll let her tell the next part of the story:

Tom Walker

Tom Walker's point of view:

"Deciding to start and finance a company based on Cheryl's talents was easy after seeing the rave reviews from her students and readers. Cheryl's enthusiasm is contagious, in person and in writing.

The natural next step was to overrule her natural modesty and focus all our marketing and our products on her experience and personality.

As an editor and having mainframe but no IBM experience, I happily functioned as the most clueless reader, asking the dumb questions and getting Cheryl to expand her explanations to accommodate people new to the subject.

Working together 24/7 since 1986 might sound difficult, but it's not for us. We're very compatible, laugh all the time, and even had identical furniture when we met!"

"Tom had just retired from working for three years for ITT, who had bought a time-sharing company he had part ownership in (Dialcomm, Inc., a forerunner of companies such as AOL). He had been in mainframe computers since 1965, but not IBM (GE, Honeywell, etc.). Since I couldn't work those hours and find time to spend with Tom, I resigned and went to work as an independent consultant and trainer."

Whilst working as an independent, Cheryl started writing articles on performance for Mainframe Journal and Technical Support Magazine. These became very popular, especially the multi-part articles on RMF and another set on SRM (Storage Resource Management). Back to Cheryl:

"In 1986, Tom decided to finance our joint company and use my name as the major item in advertising our classes.

I developed the classes, Tom reviewed and edited them, and he financed the startup. That startup included huge mailings of class advertisements, all with my name in large bold letters on the ads.

It worked, and the courses took off. We taught throughout the world - throughout the US, England, Europe, South Africa, Scandinavia, Singapore, etc. until we wanted to stop traveling. Then we simply taught classes in Florida, where we moved in 1989."

As we all know, none of us are getting any younger, and in 1991 Cheryl and Tom took the decision to change the direction of the company. Instead of her exhausting schedule traveling around the world giving classes, Cheryl began to write "Cheryl Watson's Tuning Letter". This started out as a 16 page monthly (which grew to 40-50 pages bi-monthly) newsletter devoted to: How to tune the mainframe and How to do Capacity Planning.

With Tom's help as Editor, Cheryl quickly learnt how to write well. His influence was visible in the marketing sphere too (he broke with "tradition" sending the whole of a newsletter as a flier instead of just a teaser which was normal practice at the time). The accessible style that this pair has developed has been an inspiration to my own writing style. Cheryl assures me this is Tom's influence at work and that her natural style is firmly embedded in Geek!

In 1995 Cheryl began pushing WLM (Work Load Manager) into our collective consciousnesses. She immediately realized that this was the future and that performance to specified targets could only be a good thing. Sadly we in the User Community mostly disagreed! My own reason for not liking WLM in the early days was firmly based in the fact that, with Cheryl's help, I had finally got to grips with IPS and ICS in SYS1.PARMLIB (**I**nstallation **P**erformance **S**pecifications and **I**nstallation **C**ontrol **S**pecifications) and didn't want to accept that WLM meant my hard earned knowledge was now pretty much useless!

Cheryl's reaction to this wasn't to run away and hide in a corner (as I have been known to do if things don't go the way I want – I admit it, sometimes I revert to being a 6 year old!). Oh no! Her reaction was to work with IBM to get the length of time we had to convert to WLM extended and to publish a starter set of WLM policies known as QuickStart. It has been this patient, hand-holding rather than brow-beating approach that is Cheryl's trademark and part of why she still remains so popular and relevant today.

One of the other reasons that Cheryl has maintained a highly relevant technical profile in the mainframe industry is that she is always looking for a new cause to champion. Her next target after convincing us all that WLM was achievable was to demystify the archaic performance charts that IBM uses to rate their mainframe systems.

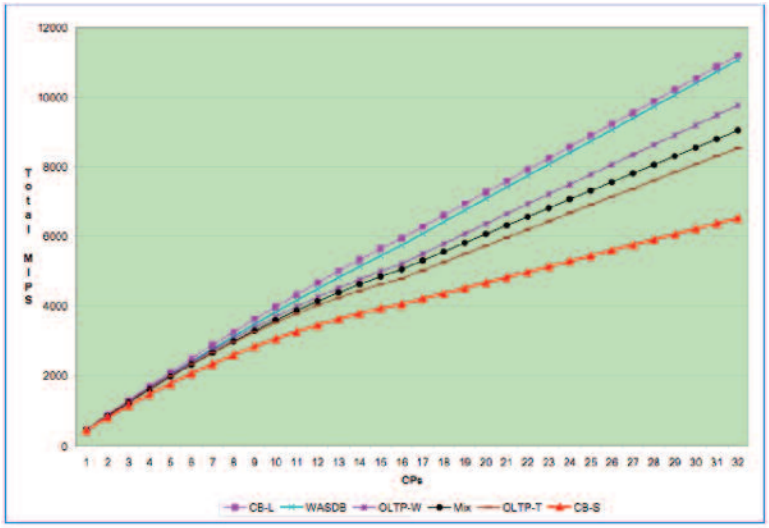
It took a great deal of study to figure out the process used by IBM and, as soon as she had done so, it was time to share with the wider community. Her first port of call was to publish her latest newsletter with details on how LSPR (**L**arge **S**ystems **P**erformance **R**eference) and processor speeds were analyzed.

BoxScore

BoxScore is an informative suite of programs, developed by Cheryl Watson that shows you the difference in CPU speed and CPU capacity after a hardware change, a software change, a change in LPAR configuration, or a tuning change.

The landscape for capacity modeling effectively changed overnight with the switch to CMOS based machines. Instead of modeling a single large processor we suddenly had to deal with multiple smaller processors which could work together.

Cheryl took the campaign slightly further than just telling people what she knew. This time she designed and wrote a product, BoxScore, as Cheryl put it: "To make sure that you are getting what you are paying for!"



BoxScore is still a vital tool for many organizations when it comes to modeling changes. It can also identify some performance type problems e.g. some jobs run **WAY LONGER** than others performing similar function.

Cheryl identified that this can sometimes be due to poor coding standards, for example, using subscripts instead of indexing when writing COBOL. Increasing your system capacity would make no difference to the performance of such examples of "bad coding".

In 1998 Cheryl suffered a heart attack after a horrific home invasion. She exercised herself back to 80% fitness within a year! On top of that, in 1999, she was awarded the CMG Michelson Award. In 2000, she and Tom married, and she became 'Cheryl Watson Walker', using Walker at home and Watson in business. I'm not sure that there is anything that can keep this Lady down for long!

Cheryl has always been passionate about the things that she believes in. Her latest campaign is based in how excited she is about z/OSMF! So what is z/OSMF?

IBM says:

"The z/OS Management Facility is intended to enable system programmers to more easily manage and administer a mainframe system by simplifying day to day operations and administration of a z/OS system. "

Cheryl Watson says:

"z/OSMF IS the future for z/OS administration!"

She acknowledges that mainframe specialists don't currently "trust" the facility but that this is because IBM made some odd decisions about its direction in the early days. Cheryl has been fighting our corner with IBM to make sure that the focus is on making the facility run better before pushing out new functionality.

She sees the slow take up of z/OSMF as being caused by the same problem as was faced by WLM in the early days. Overworked Systems Programmers don't see the immediate benefit to them. However, we have to start

thinking about the long-term future for **my** platform of choice, the IBM mainframe.

Lack of investment in new mainframe staff over the last few decades means that we have to take creative approaches to up skilling the Admins of the Future. And by Admins we are going to have to start envisioning people performing TCP/IP configuration, WLM changes, Capacity Management, software distribution, dump analysis and other, more classic ISPF (**I**nteractive **S**ystem **P**roductivity **F**acility) tasks. The wizards built into z/OSMF can simplify the tasks and reduce the learning time for new personnel.

Cheryl has kindly agreed to provide us with her personal insights into z/OSMF and the future of mainframe administration. But before we get to that, I'd like to share an extract from Cheryl Watson's first ever Tuning Letter in Appendix 1. When we were talking about which piece to include here, Cheryl and I both wanted to give the reader a feeling of the gravitas of operating in an environment that has been around since the 60s.

Cheryl explains it best: "Because the basics of MVS, the under-pinnings if you will, have not greatly changed, most of the old Tuning Letters contain articles that are relevant in today's environment (and are still available). These two articles from my first Tuning Letter in January 1991 illustrate that nicely. The MVS Review relating to program search is still the same, with the slight addition of a new PROGxx member. The Reducing ISPF Response Time is still valid, but less important because the default IBM distribution now uses the preferred placement of ISPF LPA (**L**ink **P**ack **A**rea) modules."

The extract is available in Appendix 1 whilst the full Tuning Letter can also be found here www.watsonwalker.com/JAN91.pdf).

So where does this leave us? We have more and more mainframes being used, but are left with a small, and shrinking, number of staff to look after them. Ooops!

But don't panic! IBM is coming to our aid with z/OSMF, making the everyday tasks (and some of the not so every day) more accessible to other staff. We can concentrate our skills on the more technical work. z/OSMF can help to free up our time so that we can get our organization's 'Big Data' stored securely, whilst maintaining high availability, but still passing those all-important audits. This is what we all want after all. Appendix 2 allows Cheryl to go into more detail about what z/OSMF is and how it can help you.

Anything that allows us to improve the performance of our platform is important. It means we can get to work on those tasks that have been sitting on the back burner and are still waiting now. We may now have the chance and "z/Auditing Essentials VOLUME 2 – Auditing z/OS" will help you!

I hope you enjoy Cheryl's new paper (in Appendix 2) as much as I did.

Remember: Civilization runs on MVS – Julie-Ann
(Shamelessly adapted from one of Bob Rogers' old tag lines)

Roll call of my Mainframe Heroes

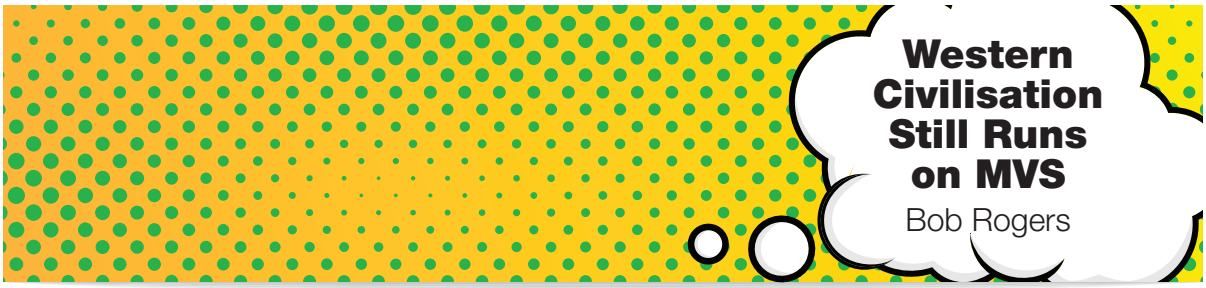
There is not enough space here to give credit to all of the individuals who have made such a difference to the industry I work in but this should give you enough to be able to Google them yourself.

In alphabetical order…

Barry Merrill	*who… was a pioneer in performance management*
Barry Schrager	*who… introduced us to Data Security*
Ben Riggins	*who… gave us CICS*
Bob Rogers	*who… gave us ASM in MVS and then 64bit in z/OS amongst other things*
Cheryl Watson	*who… introduced us to Performance and Capacity Management*
Eldon Worley	*who… gave us RACF*
Gene Amdahl	*who… was the Chief Architect for S/360*
Karl-Heinz Strassemeyer	*who… brought Linux to System z*
Kenneth Kolence	*who… co-founded Boole and Babbage*
Mike Cowlishaw	*who… brought Rexx into the world*
Pete Clarke	*who… saved VSE!*
Peter Relson	*who… brought us the IBM Health Checker for z/OS*

Check out the "Mainframe Hall of Fame" for more influential individuals who have helped to shape my platform of choice.

http://www.mainframezone.com/mainframe-hall-of-fame



Foreword

I've been a "Mainframe Evangelist" for the last 30 years. For most of that time I've felt like a lone voice in the wilderness. More or less everyone you meet in the IT industry has an opinion - even if that opinion takes no account of facts - on the level of deadness of the platform. These people, the ones who lead the charge in calling our work "legacy", generally haven't worked with IBM mainframes. They see the fact that the platform has been around for so long as a bad thing.

We who work with IBM mainframes choose to differ!

The latest generation of mainframe combines what the uninformed call "legacy" with a number of bleeding edge technologies that allow users to move their legacy applications seamlessly into the future, doing more with less and providing ever more complex solutions to a diverse user base.

From banking through to governments and space exploration through to online supermarket shopping, the mainframe has had a massive influence over our lives and will continue to do so. But I won't argue that we've done enough to convince the rest of the world of that! There has been massive under-investment in the platform. Well, "everyone" just "knew" that it was a "dead platform", didn't they?

One of the results of this under-investment has been the gradual loss of both skilled people and the knowledge that they have brought to the platform. Back in 2009 I started working on a project to try to document some of the required skills for running a mainframe in the 21st century. The project was envisioned by my colleagues at New Era Software Inc, and their vision is for us to provide a repository of knowledge that everyone can use.

The first part of the project looked at the Essentials of Auditing CICS (the z/OS based transaction server which sits behind the majority of ATMs). The book (CICS Essentials: Auditing CICS - A Beginner's Guide) can also be used to secure CICS regions. The informative but casual style made the publication very popular.

Next to be published was Volume 1 of z/Auditing Essentials: zEnterprise Hardware - An Introduction for Auditors. This book documented the "Front Doors" that any security strategy for System z must take into consideration.

The latest part of the project is Volume 2 of z/Auditing Essentials and will be looking at Managing the 21st Century Mainframe. What new challenges await the IT departments that we work within? Well, you can bet your bottom dollar on us needing to take Big Data seriously. And I wouldn't be surprised if performance management were to become a rapidly changing discipline.

What we have done is to start this latest volume off with a series of interviews with the people who are

going to be amongst the most influential as the mainframe progresses to the next stage of computing. We started with an interview with Barry Schrager about his experiences, and influence in the security side of mainframe computing and then went on to talk to Cheryl Watson, the go to guru for all matters relating to performance on my platform of choice. This work chats to Bob Rogers about his life in the MVS Development team and where he thinks we are going next.

I know I am a better MVS Systems Programmer because of my friendship with Bob (and his willingness to chat about the internals of the operating system I chose to base my career on) and so it is with great pleasure that I present the latest installment of the z/Auditing Essentials series.

Julie-Ann Williams - April 2013

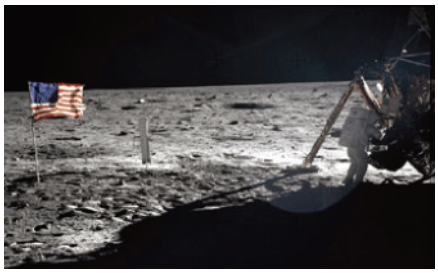
Western Civilisation Still Runs on MVS

This is the third paper introducing a “Who’s Who” of some of the brightest, and yet under-sung stars of our mainframe generation. These folks have kindly agreed to be interviewed and will be providing their input to the new book, “z/Auditing Essentials VOLUME 2 – Auditing z/OS” that follows on from the acclaimed “z/Auditing Essentials VOLUME 1 - zEnterprise Hardware - An Introduction for Auditors”.

Bob Rogers. On top of Sydney Harbour Bridge!

I have mentioned before how lucky I am to be able to work with so many of my heroes but this time, in the case of Bob Rogers, it feels more like interviewing a minor deity!

I first met Bob at a G.U.I.D.E. conference in Brighton, England in the early 1990s (**G**uidance for **U**sers of **I**ntegrated **D**ata processing **E**quipment was the leading European IBM User Group at the time which later merged with SHARE in Europe to become GSE or G.U.I.D.E. SHARE Europe). It was a time where simultaneous translation to multiple other languages was still being done at technical conferences and Bob was getting through three teams of translators for each one hour presentation he did! They simply couldn’t keep up with the number of words that this engaging New Yorker was capable of fitting into each minute. He was one of the first people I ever met who had a positive message to share about mainframes.



Born in Dutchess County, NY, Bob puts me in mind of a nerdy version of George Clooney (it’s something about the eyes!). I remember he once told me that he had tried to teach himself ancient Greek just so that he could read Homer’s Iliad un-translated! But the reason for me “worshipping” the man starts way back in the mists of time…

It was 1969 and the year that IBM computers helped NASA to put the first men on the moon, when Bob got his first job as a Computer Operator at the IBM Poughkeepsie Center, where MVS was being developed. He continued to study for his B.A. in mathematics, from Marist College, whilst working for IBM.

I’ve often wondered if Bob would have been diagnosed with ADHD if he was growing up now. Or at least the hyperactivity part of it! I’ve never known Bob to voluntarily take a break but I read a quote in an interview with him when NaSPA (Networks and Systems Professional Association) included him in their Mainframe Hall of Fame which I think gives a little insight into this. In response to a question about the length of his working week, he said: “Ever since I was an operator, it has been difficult for me to tell when I'm working and when I'm not.” By the way, he’s also in the Enterprise Systems Magazine Mainframe Hall of Fame – one of the few people to be recognized on both lists.

It reminds me of an old proverb, sometimes attributed to Confucius that says - “Find a job you love and you’ll never work another day in your life”. Well Bob certainly enjoyed his job and working within the environment where MVS was being developed led to some unique circumstances. For example, Bob realized he had access to not only documentation but people who were happy to help him to learn. So he started learning coding languages and would use machine idle time to see what he could make a

James Bond doing an HSM recycle?

mainframe do.

One of his "projects" (another of which would eventually lead to his Boss removing the terminal from Bob's office!) was to try to make a mainframe behave like a "TV computer". You know the type of thing where a character is using a computer and you know it because you can see a tape drive spooling backwards and forwards?

Real life computers don't behave like this. Let's face it, if a program read/wrote to tape that often it would have been re-written to use disk files! So Bob had to come up with an idea to create a whole heap of tape spooling activity. He did it with a program to compute prime numbers. With a bunch of known prime numbers pre-stored on the tape, whenever the algorithm looked at its list of already proven prime numbers, the tape would move creating the almost continuous movement that was the effect Bob was looking for.

__R__esource __A__ccess __C__ontrol __F__acility

RACF®, as the z/OS security manager, is responsible for making all access control decisions in z/OS.

- *Identify and authenticate users*
- *Authorize users to access*

So when he achieved his B.A. 2 years later, it was no real surprise that he was sent to the IBM programming school. After that Bob went straight into a Junior Programmer role at IBM and has never looked back.

After working for a few years on projects that got canned, the first real project that Bob was involved in (he joined the team in 1974) was as a development programmer on the RACF product team. He designed and implemented the database component for release 2 as well as the initialization component for release 3. He says:

"While in this role I implemented database capability for duplicate data set names and multi-volume tape data sets. I also implemented support for multiple RACF data sets for release 3 and co-wrote the Split/Merge Utility (ICHUT400) in a lull while design issues were resolved."

So now we know who to blame!

Just 2 examples of Bob's code:

- *XA version of the IPL bootstrap module which is the first software to get control from the hardware at IPL time aka the mainframe version of CTRL, ALT, DEL;*
- *Initialization code for the Real Storage Manager (RSM) and Virtual Storage Manager (VSM) components which were fundamental to the very early virtualization capabilities offered by IBM mainframes.*

But it was his next role that took him into the area that I most think of Bob in. It was 1976 when he became the Component Designer for MVS/XA system initialization components (IPL/NIP) and the Contents Supervisor component. These are parts that make up the low level operating system functionality (they are often in direct communication with the hardware) and many of the components written at that time are still in z/OS today. His approach to the role of Component Designer meant that Bob always tried to make designs as "future-proof" as possible, hence the very long life of such a lot of his design ideas!

I asked Bob about this time in his life and his typically, unassuming response was: "I also wrote a lot of the code." That code included some of the most deeply technical elements of my operating system of choice.

His future looked set to be one of those IBM Lab guys who are rarely, if ever, allowed out to speak to customers. Fortunately for us, fate had different plans for Bob.

I got my first job as an MVS Systems Programmer in the early 1980s around the same time that Bob started working as a kind of "Team Leader of Team Leaders" for all of the serviceability components in MVS (basically the operating system).

When I started my first job I was warned by “the people who knew” that I was wasting my time getting into a dead end career because “everyone” knew that mainframes would be gone within the next 3-5 years.

I can’t imagine how frustrating it must have been for Bob to be so central to the production of this “dead platform”. Fortunately, Bob has a couple of stories that highlight very clearly how he got over this and just how he felt at the time!

This first story is about Bob’s inaugural presentation to SHARE. He had been asked to present on “the serviceability stuff in MVS/ESA”, an incredibly dry, technical topic at the best of times. The IBM process for a member of staff to present in public was extremely rigid at the time, culminating in practicing the presentation in a cigar-smoke filled cubical, with multiple Senior IBM Managers in attendance. I can’t think of many more intimidating situations. But Bob got through his practice run without major incident having memorized the whole presentation.

Quite clearly, presenting to a handful of IBM’s Senior Managers isn’t the same as standing in front of a 500 person audience. Particularly when you consider that you have no control over when your acetate/foils are changed on the overhead projector because you are standing at a podium on a stage with someone else controlling the projector. That’s why we used to have to write scripts for presentations.

And the one thing that the formalized process at IBM didn’t allow for was audience research. Who were these 500 people? What kind of language did they use? Were they friendly? Even, what clothes would they be wearing? As a speaker, knowing your audience can have a massive impact on how comfortable you are when giving your presentation. The converse also holds true (the less you know about the audience the more nervous you end up)!

One of the problems that many of us folks who got into computing at a fairly early stage have is, getting over the nagging worry that we don’t really know what we’re doing and that someone is bound to find us out eventually! There were no computer science degree courses back then. Heck, PCs hadn’t even been invented yet!

Fortunately, the members of SHARE are lovely people who are genuinely interested in technical presentations. And with Bob’s session being at the end of the conference, he managed to lay his final fears to rest. It went very well and Bob’s career as a speaker was out of the starting blocks.

The lessons Bob took away from the whole thing were 1) User Groups are generally populated by nice people who are only scary in very limited circumstances and 2) Never try to memorize an entire presentation, always give yourself room to adapt the session to your audience (and your own concentration levels!).

Bob’s speaking abilities (not to mention his popularity with customers) saw him becoming deeply involved with the sales process for MVS/ESA as well as retaining responsibility for the design work. MVS/ESA was a complicated and technical sell and Bob’s style was perfectly suited to the job. In parallel to this (and his regular job) he also became the IBM Representative to the old GUIDE User Group. Run along the same lines as G.U.I.D.E. in Europe, this was always the “technical” conference MVS Systems Programmers world-wide attended if someone in their group had already gone to SHARE.

You might think his additional duties would prevent Bob from remaining so involved with the design and development of MVS but this was not the case. During the same period he also:

“Created the overall design of how the components would support the ESA architecture and directed the development teams for three releases. Defined the interaction between ESTAEs and Automatic Recovery Routines (ARRs) and the relationship between recovery routines and mainline code invoked by hardware-assisted linkage (i.e. Stacking Program Call) and system-assisted linkage (i.e. the LINK and SYNCH macros); Personally wrote code to improve SVC Dump elapsed time; and rewrote ESTAE to solve a horrible serialization problem.”

All on a mainframe!

Bob continued to progress up the technical side of IBM's dual pyramid, spending 3 years in the late 1980s as an MVS Performance designer and analyst. During that time he worked very closely with both the Labs and customers to analyze hardware and software performance data, identifying problems or opportunities in the latest operating systems release. He used all of this information to simply: "produce designs that performed better".

The second story is also about a presentation. This time, Bob had been asked to do a 30 minute informal session on any topic. He chose to show how integral the mainframe is to so many aspects of everyday life. We were a long way into being told that it was a dead platform and this story took place shortly before IBM started their famous series of double page print advertisements singing the praises of the mainframe. In fact, this story might even be considered to be the catalyst for that campaign.

Having established that User Groups mostly contain lovely people, Bob went to them for help. He asked for stories about "really cool" things people were doing with MVS. He got a ton of them! From all ATM transactions eventually ending up in a CICS region on MVS, through to the air cooled 4381 being shipped to the Theater of War during Desert Storm with stop offs at all 800 numbers being generated using MVS and even jet liners sending all of their air-born statistics to an IMS database running on MVS somewhere (which explains some of those odd kinks in flight paths)! All of this on top of the normal, boring, "legacy" stuff.

All of these really cool uses for the mainframe convinced Bob that he was right - the mainframe simply **CAN'T** disappear in a few years. So he went about putting together some thoughts for his talk to this internal meeting of the MVS developers.

On the morning of the presentation it was raining. You know those Hudson Valley rains that seem to permeate through to your very bones? Quite frankly, the weather was depressing. To kill time until it was time to drive to the auditorium, Bob turned on the TV (something he rarely does). Good Morning America was on, and he was just in time to catch a discussion on how and why the mainframe was destined to die! Having known Bob for quite some time now, and given a similar set of circumstances, I would not be joining him for a cheerful breakfast! In fact, he used the word "incensed" to describe his feelings just before the presentation.

Bob has a really contagious presentation style that helps to bring an audience along with him for the ride leaving them feeling breathless but committed to whatever cause has been the topic of the day. He also has the most complete understanding of the way that MVS works that you'd ever wish to come across. He'd been central to making sure that IBM's customers were getting what they needed out of the operating system so he really understood how to explain the importance of MVS. And to do it with passion - something not often seen in our industry!

The basis of his argument was that MVS can't die all that quickly. It is far too deeply imbedded in so many parts of everyday life that it would be impossible to remove. Bob told the audience about a heated discussion that he had with a colleague who was a Macintosh operating system specialist. Her point of view was that the mainframe was just a relic of a former time and that Macs would take over all of the work done on mainframes within the next 2-3 years.

That may have been her first mistake! Bob calmly agreed with her that it would be big news if all of the

Mac systems around the world stopped working. In fact, he agreed, that it would probably make front page news of the New York Times. However, he went on to tell her that he felt if all of the mainframes in the world stopped, it wouldn't be reported (because the paper could not be published, distributed and sold!) and that Western Civilization would crumble within 3 to 5 days.

The evidence is incredibly strong for this position. Almost all banks globally run mainframes. We had a glimpse of what levels of chaos can be provoked by a failure in a single software package on a mainframe just last year in Europe. The "Denial of Service" incident at RBS in 2012 was caused by accident. But the weeks of chaos, which included millions of people not being able to move money or pay bills, was just as damaging as it would have been from deliberate damage to the mainframe environment - but the problem would have taken much longer to fix on any distributed platform. The customer compensation costs alone ran to £125 (c$190) million!

And by the way, it's not just banks. Global users of mainframes still include airlines, and telecoms, and government, and supermarkets, and big manufacturing organizations. All of which go to building "Western Civilisation".

So it turns out that incensed and armed with a ton of facts is a really good way to go into a presentation to defend the honor of the mainframe! In fact it went over so well that notes were being sent internally, to the division president, telling him that he absolutely had to hear the message that Bob was sharing with the world. However indirectly, I like to think that it was Bob's enthusiasm that led to the series of double page adverts from IBM hammering home the relevance of the mainframe.

So back to Bob's career progress - In 1991 Bob became an Advocate for Independent Software Vendors aka ISVs. The operating system was becoming ever more complex and IBM had acknowledged that there were some areas that they weren't going to develop. These areas were ripe for others to simplify for the customers though. Those others were the ISVs. And whilst there have always been ISVs, this is the first time that IBM had given them special help.

Bob's role was to educate ISVs on what was coming in future hardware and software so that their products could support the IBM products when they became available. He worked cooperatively with ISVs on mutual customer problems and requirements. I was working for an ISV at the time and it made a huge difference to our customers **and** to us.

As Bob said in the NaSPA Mainframe Hall of Fame interview, "I've spent a disproportionate amount of time working on taking the whole mainframe environment and moving it forward to a new addressing architecture." He's quite right! For 10 years of his IBM career (1997-2007) Bob was OS/390 and z/OS Software Architect and Designer of the evolution of the platform into the 64-bit world. He was **THE** MVS expert for this important aspect and directed the development teams for a few releases to execute his 64-bit roll-out.

Among the highlights of his time in this role are included some massively important projects like:

- Created the system level design for OS/390 support for z/Architecture incorporating 64 bit real **and** 64 bit virtual memory;
- Personally designed and implemented the support for up to 32 processors in a single z/OS image;
- Lead the design for the z/OS exploitation of the zAAP and zIIP specialty engines.

Bob was given the title of IBM Distinguished Engineer in 2007. There are only ever a handful of these guys around at any one time and they are the acknowledged experts in their fields. Whilst the rest of us see this as an astounding achievement, Bob really saw it as a bit of a nuisance that got in the way of him doing some of the stuff he loved. It did give him two new interests though: he became the IBM System z Strategist for hardware/software synergy, responsible for fostering communication between the two communities to achieve optimal performance up and down the hardware and z/OS software stack; he also started to get involved with the next generation, explaining exactly how a modern mainframe works.

Underlying his technical knowledge of System z, Bob is named on eight separate technical patents associated with his work at this time (with dozens more still pending).

I love hearing Bob talk about things he is passionate about and his discussion with me about patent number US 8276151, on which he is co-author along with Greg A Dyck, Mark S Farrell, Charles W Gainey, Jeffrey P Kubala, and Mark A Wisniewski, was one of those classic conversations.

One of the things that I love about our friendship is that Bob always starts from the position that I simply must **ALREADY** understand System z millicode (the clever bits of code that interface directly with hardware functionality that can be exploited to help provide performance improvements). I've tried to adjust the language to a slightly less technical arcana in the following description.

So, this particular patent deals with determining whether or not a logical processor which z/OS believes to be running work is actually dispatched by PR/SM and executing instructions. This is important in the situation where one logical processor wants to obtain a system spin lock and some other logical processor already holds it.

When a required system spin lock is not available, z/OS must spin (that is, loop doing nothing but checking to see if the lock has become free) and is unable to do productive work until the lock is obtained. This can be quite wasteful, but becomes much more wasteful if the logical processor holding the lock is not even dispatched by PR/SM. If that's the case, then the logical processor holding the lock makes no progress towards freeing the lock for a relatively long time (say, a dozen milliseconds) until it is again its turn to be dispatched by PR/SM.

Several attempts were made to solve this problem. The first attempt was for a logical processor to yield the remainder of its PR/SM time-slice hoping that the logical processor holding the lock will have released it by the time that the current logical processor is re-dispatched. This wasn't optimal since it introduced a lot of PR/SM overhead in the re-dispatching of the logical processor that had yielded its time-slice. The next attempt was to spin for a while to give the holding processor some time to free the lock before yielding. This was very effective if the holding processor was running but increased the waste if it was not.

To Bob, the key was determining whether the holding processor was running or not, and this was the gist of the invention. An instruction was devised that allowed z/OS to ask the machine millicode if the other logical processor is currently running. Now the logic was simply, if the holding processor is running, keep spinning; but if it isn't, then yield the rest of the time-slice to give the holding processor an opportunity to get dispatched by PR/SM and free up the lock. The problem was solved and eventually the patent was issued. It is solutions like this that make CPU virtualization on System z superior to virtualization on other platforms.

Bob continues with his interest in making sure that optimum performance is squeezed from System z hardware today. And retirement in November 2012 doesn't seem to be slowing him down any.

When I asked about his entry into the world of retirement he told me that he had to buy a laptop for the first time after he retired! He also said: "While under a non-compete agreement, I am doing conference presentations for some ISVs, writing articles for IBM Systems Magazine, and seeking interesting work which does not compete with my former employer."

Some of that work includes helping to publicize the fact that we are likely to run out of technological advances in hardware as we approach the limits imposed by the laws of physics. In fact, Bob's first public engagement post-retirement was as a Speaker at SHARE San Francisco in February 2013.

Working with one of the organizations that he met through his work with the ISVs, Bob was sponsored

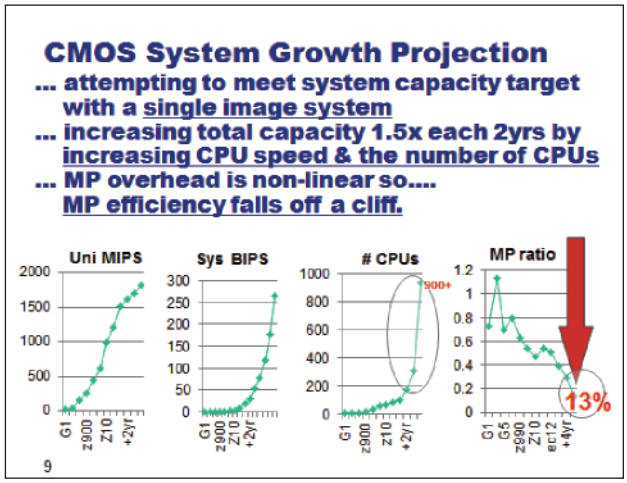
by INNOVATION Data Processing to produce a white paper and associated presentation explaining how we got to where we are now with regard to mainframe capacity and what shape the future is likely to take. The full paper is available from the INNOVATION Data Processing web site here:

(http://www.fdr.com/pdf/LNL_SHAREsanFran_BobRogers.pdf)

Since the introduction of CMOS processors back in 1994, we in the mainframe world have had the luxury of not having to worry too much about application performance. Anyone who was reaching the limits of the capacity on the current machine could be assured that waiting for the next generation of mainframe would resolve the problem.

This is effectively Moore's Law at work. As Bob says in the introduction to the white paper he wrote with INNOVATION Data Processing:

"Corporations, governments and other large enterprises always have and always will have a nearly insatiable demand for more mainframe computing capacity. Over the last four decades advances in hardware chip technology met the lion's share of this demand. However, we are entering an era when hardware advancement is most likely not going to continue at the same pace. So now is a perfect time to review other techniques for growing capacity like multiprocessing and Sysplex clustering; and to prognosticate on how we can continue to grow mainframe capacity, without benefit of the large contributions from chip technology that we are so accustom to. Your future will include expanding the use of MP and Sysplex, but you'll also have to be ready for some entirely new technologies that are likely to be introduced into your mainframe environment."



Moore's Law is the observation that over the history of computing hardware, the number of components on integrated circuits doubles approximately every two years.

There are 3 main "barriers" that stop engineers from being able to continue to make CMOS processors faster. Memory access times aren't getting any faster so the CPUs spend more time waiting for data, power density limits as more, higher frequency circuits are packed into smaller chips and frequency limits where increasing pipelining length has also come to its limit.

Actually, hardware engineers are predicting that growth in capacity will slow to single digit percentage rises. And Bob's best guess is that these problems are going to be biting us by 2020. He believes that "by the time we are 6 years out, adding more processors adds almost no additional capacity".

The new EC12 - Still think there's no future in it?

I would **very** much like to emphasize here that the precise point where any particular enterprise will be impacted depends on its size and rate of growth. But this is not just a mainframe problem and every industry platform is affected. There are ways to survive this "crisis", but some of the options available are not classically considered attractive ones. For example the Board might need some convincing that your applications need to be rewritten to allow them to take advantage of parallel processing when in the past they have always simply bought a bigger box! But, Bob suggests that if you're running on z/OS with the right middleware, Parallel Sysplex may yet let you dodge that bullet.

IBM continues to strive for more capacity and is coming up with many clever tricks (such as "cracking" where complex instructions are broken down into a sequence of more simple ones for speedier processing) to keep the problem in the background. But the drive towards ever bigger and more

complex systems seems never-ending. Analytics and Big Data can only drive us further towards the cliff if we continue to ignore that the problem even exists.

Bob Rogers is not the only person in the world talking about the problems we face with regard to handling increasing demand, but he is one of the most infectious! His passion and obvious technical stand-point rarely fail to sweep an audience along with him. As one of the earliest mainframe evangelists, thank heavens Bob uses his “power” for good.

He has a very quiet looking resumé, having only ever worked at IBM. And 43 years with one employer is extremely unusual in the 21st century! But this doesn’t adequately reflect the deeply technical career that has also taken him around the world many times as a compelling speaker.

He describes himself as having both a “Deep understanding of z/Architecture and System z instruction processors and memory nest” and a “Better than average speaking ability”. Now that’s the Bob Rogers I’ve worshipped for years! ;-)

Remember: Civilization still runs on MVS – Julie-Ann
(Shamelessly adapted from one of Bob’s old tag lines)

Appendix

A Perspective on the IBM System z Mainframe Capacity delivery… past and future

Bob Rogers
SHARE San Francisco February 2013
Sponsored by INNOVATION Data Processing

Bob.Rogers@ColossusConsulting.com

A Perspective
IBM System z Mainframe
Capacity delivery…
past and future

Bob Rogers
SHARE San Francisco February 2013
Sponsored by INNOVATION Data Processing

Bob.Rogers@ColossusConsulting.com

Overview

Large and larger enterprises ==> growing demand
Cloud, Big Data, Analytics ==> insatiably growing demand

Forty years of hardware advancement ==> met demand
Slowing hardware advancement ==> hard to meet demand

Multiprocessing & Sysplex Clustering ==> help meet demand
Future… new technologies ==> needed to meet new demand

2

Abstract

Corporations, governments and other large enterprises always have and always will have a nearly insatiable demand for more mainframe computing capacity. Over the last four decades advances in hardware chip technology met the lion's share of this demand. However, we are entering an era when hardware advancement is most likely not going to continue at the same pace. So now is a perfect time to review other techniques for growing capacity like multiprocessing and Sysplex clustering; and to prognosticate on how we can continue to grow mainframe capacity, without benefit of the large contributions from chip technology that we are so accustom to. Your future will include expanding the use of MP and Sysplex, but you'll also have to be ready for some entirely new technologies that are likely to be introduced into your mainframe environment.

Trademarks and statements:
IBM, System z, z900, z990, z9, z10, z196, EC12 and z/OS are trademarks or registered trademarks of International Business Machines Corporation. All other service marks, trademarks or registered trademarks are the property of their respective owners.

http://www.fdr.com/pdf/LNL_SHAREsanFran_BobRogers.pdf

Viewpoints

Stu Henderson
and Mark Hahn

Foreword

In addition to the Superheroes profiled in this book, we want to acknowledge other contributors to The z Exchange that we consider Superheroes as well.

Stu Henderson of the Henderson Group and Mark Hahn, now retired from IBM Corporation, offered insights into RACF that were included in zAuditing Essentials, Volume 2 – Taming RACF – SETROPTS.

The zAuditing Essentials series of eBooks have been published by NewEra Software and are available free of charge. They include:

- zAuditing Essentials, Volume 1 – zEnterprise Hardware
- zAuditing Essentials, Volume 2 – Taming RACF – SETROPTS
- zAuditing Essentials, Volume 2 – Mastering CA ACF2 - GSO
- zAuditing Essentials, Volume 2 – Controlling CA Top Secret

You can download all eBooks at http://www.newera-info.com/eBooks.html

The following pages are Viewpoints from Stu Henderson and Mark Hahn.

How to Determine the "Correct" SETR Option Settings

By Stu Henderson

Stu Henderson

You've probably figured out by now that there is no single set of correct SETR option settings. If that were the case, IBM would make them all be the default. This appendix shows both auditors and security administrators how to go about determining and documenting what RACF settings make sense in a given installation.

So how do you know whether the options you see in an audit, or the options you set yourself, are correct? The body of this book gives you a large amount of information, but quite intelligently doesn't recommend a single set of options.

(If you want to use the government STIGs as an absolute audit checklist, please remember that the "G" is "STIG" stands for "Guide", not for "Regulation".) So how do you know what's right? And how do you identify possible benefits that don't relate directly to reducing risk?

The answer, as we all learned in Auditing 101, is to start with the risk assessment. No security audit finding is meaningful unless it does one of two things:

1) Demonstrate violation of a law, regulation, policy, or standard
2) Demonstrate significant risk

So if you are considering for example whether BATCHALLRACF should be active or not, there is no law. But if it is not active, you know that it becomes possible for a batch job to execute without having a RACF userid, that is, without being blessed by RACF. ("So what?", I hope you're asking now.)

To answer "So what?", evaluate the harm that might result from this. You realize that if BATCHALLRACF is not active, then CICS, MQ, RJE, NJE, and FTP can be used to submit batch jobs without a userid. And if a userid logs onto TSO without providing a RACF userid and password, that user can also submit batch jobs without a RACF userid. And so can any started task which starts without a RACF userid.

Digging deeper to understand the risk, you understand that if any dataset or resource profile has an inappropriate UACC, then a batch job without a RACF userid would be able to access it, putting data and resources at risk.

Plus, if some employee can execute batch jobs without RACF userids, she could start a computer service bureau, charging customers a fee to use her employer's mainframe. (It's been done.)

Now some might say that if you see BATCHALLRACF is not active, you can still have good security. You just have to plug software like RJE, NJE, and FTP, and make sure that every TSO user and started task runs under a RACF userid. Oh, and be sure that every dataset and resource profile has an "appropriate" UACC.

But isn't it easier just to flip the switch to make BATCHALLRACF active? Doesn't that make it easier for the auditor to say "controls appear adequate."? Doesn't that make it easier for the RACF administrator to sleep at night, without having to check all those other places?

So some basic questions to be addressed in the risk assessment for SETROPTS options include:

- What does this make possible?
- What harm could result if that were to happen?

- How easy is it to prevent this from happening?

Well then, what about XBMALLRACF? Won't a risk assessment show in similar fashion that effective security is easier to demonstrate if this is active? That depends of course on what effect XBMALLRACF has, which is why the body of this book can prove useful. As you've probably read above, XBMALLRACF has an effect similar to BATCHALLRACF, but only for joblets processed by the JES eXecution Batch Monitor feature (XBM).

This XBM feature is not used at all in most z/OS shops. (It is a performance improvement feature useful in situations like universities, where there are thousands of compile jobs every day. Since there is significant overhead in starting a batch job, this feature combines large numbers of near-identical batch jobs into a single batch job. The near-identical batch jobs are then considered "joblets" contained within the single batch job. If XBMALLRACF is active, then any joblet without a valid RACF userid is failed.)

So if an installation doesn't use XBM, then why do we need to activate XBMALLRACF? The answer is, XBMALLRACF not being active presents little risk in this case, and may likely not represent an audit finding.

On the other hand, if XBM is not being used, then making XBMALLRACF active should have no effect. It also would provide protection against the JES system programmer activating XBM and forgetting to tell the RACF administrator.

So maybe BATCHALLRACF has a place in the audit report, and XBMALLRACF has less importance, while still being desirable.

The point of this whole discussion is that no checklist of someone else's recommendations can be more than a guide of what to think about. A proper risk assessment is needed to make the determination of how SETR options should be set, whether for an audit or for a RACF administrator's self-assessment. And the recommendations should consider the cost and effort to implement.

The remainder of this appendix provides some items to consider when conducting full risk assessments for SETR, starting with some other options that provide for comprehensive protection: PROTECTALL, TAPEDSN, and ERASE-ON-SCRATCH. This is followed by more generalized advice on effective auditing of SETROPTS options.

PROTECTALL

PROTECTALL is a switch that, when active in FAIL mode, prevents access to any dataset whose dsname is not covered by a RACF dataset profile. (The one exception to this is users with the SPECIAL attribute who are able to access such datasets when PROTECTALL is active.)

Note that PROTECTALL applies only to datasets. If you want the same coverage for a resource class, consider creating a "backstop" rule, one named ** which matches every name in the class. You can specify AUDIT(ALL) on the backstop rule, so that every access it allows is logged to SMF. You can make the UACC either ALTER or NONE, as you see fit. (Please do not create a backstop rule for the FACILITY class, or for any other class where security decisions are based on the existence of a given rule. JES uses the existence of certain rules in the FACILITY class to determine what kind of security handshake to have with other JES nodes. A backstop rule in this class would cause JES to think that the certain rules exist.)

To conduct the risk assessment, you ask what harm could result from having PROTECTALL inactive? Thinking this through, you realize that users could create datasets with non-standard names, making it difficult for anyone to tell what the dataset is. Further risk results from the possibility that a dataset with a non-standard dsname could make its way into production JCL, leaving production data without RACF protection.

When you consider the effort to implement PROTECTALL, you realize that it is almost trivial. (Activating PROTECTALL is much eaiser because of its WARNING mode, use of generic profiles, and the fact that catalog management controls what high level qualifiers are allowed.) By enforcing dataset naming standards, PROTECTALL gives the added benefit of making it clear for any dataset name whether it is test, production, system, or personal, and who it belongs to.

Understanding the risks, costs, and side benefits of options such as PROTECTALL makes it easier to audit, and results in better security decisions.

TAPEDSN

You may have thought that turning on tape dataset protection with SETR

TAPEDSN was good enough to protect tape datasets and should always be required by the audit. But an effective risk assessment will show you that this is not always the case. Here are some reasons TAPEDSN isn't always enough, followed by two, often better, ways to provide protection. The ways involve a member of parmlib named DEVSUPxx, as well as your tape management software.

But first here are some commonly ignored problems in protecting tapes:

Problem 1: If you turn on **PROTECTALL** and **TAPEDSN**, and then someone wants to process a foreign (that is, from some other data center) tape, you have to give them a way around RACF. Some people write RACF exits, others use BLP (Bypass Label Processing); still others give out the SPECIAL privilege. None of these is ideal.

Problem 2: Tape labels (the records on the tape that tell you the dsname of the dataset and the volser of the tape cartridge) carry only the right-most 17 characters of the dsname. So unless you store the full 44 character dsname of the tape dataset somewhere, there is a security hole. There are two places you can store the full 44 character name to plug this hole: the TAPEVOL resource class in RACF (which almost no one uses) OR your Tape Management Software (**TMS**).

Another way to provide better protection involves four values you can set in parmlib member DEVSUPxx. They are:

- **TAPEAUTHDSN** (= YES or NO, defaults to NO)
- **TAPEAUT1H F** (= YES or NO, defaults to NO)
- **TAPEAUTHCR4** (= ALLOW or FAIL, defaults to FAIL)
- **TAPEAUTHCR8** (= FAIL or WARN, defaults to FAIL)

The latter two only apply to RACF calls caused by the first two.

TAPEAUTHDSN tells the system whether to call RACF for tape datasets (similar to SETR TAPEDSN, but doesn't use TAPEVOL records).

TAPEAUTHF1 can be used to tell your Tape Management Software to call RACF for every dataset on a cartridge instead of just the dataset you are reading. This gives extra protection against someone authorized to one dataset on a tape using that authorization to access other datasets on the same tape.

TAPEAUTHRC4 tells the system what to do if RACF is called by either TAPEAUTHDSN or TAPEAUTHF1 and RACF indicates "no dataset rule matches this dsname." This is a way to bypass PROTECTALL just for tapes

TAPEAUTHRC8 tells the system what to do if RACF is called either by TAPEAUTHDSN or TAPEAUTHF1 and RACF says to fail the request. This is like a warning mode, but just for tape datasets.

So suppose that you want to protect tape datasets using your regular dataset rules in RACF, but not using the TAPEVOL resource class. And you have **PROTECTALL** turned on. But for foreign tapes, that is, tapes from other data centers with dsnames that don't match your naming standards and don't have RACF rules, you want to allow access to anyone. You could turn off TAPEDSN (SETR NOTAPEDSN) which would de-activate the regular call to RACF for tape datasets. And then have the DEVSUPxx member of parmlib specify **TAPEAUTHDSN=YES** and **TAPEAUTHRC4=ALLOW**.

Now the system will call RACF for tape datasets, but will allow any tape dataset access request that has no matching dataset rule in RACF. Ineffect, this turns off PROTECTALL, but just for tape datasets, which might be exactly what you want to do.

Please note also that your Tape Management Software can call RACF to provide greater functionality, and/

or to ask if a user is allowed to access a specified dataset.

The TMS can also use RACF to:

- Control who can BLP (Bypass Label Processing)
- Solve the residual data problem for tape datasets (Erase-On-Scratch, described in the next section, only works for disk datasets, not for tape)
- Prevent a user with access to a tape dataset from having the same access to every dataset on the cartridge

As with other options, to evaluate TAPEDSN in RACF, you need to understand the risks, be aware of all the methods used to manage them, and determine whether the risk is reduced to a reasonable level. An auditor who sees that TAPEDSN is not active in the SETR LIST output should not automatically make this an audit finding.

ERASE-ON-SCRATCH

ERASE-ON-SCRATCH, or EOS, is an important feature which is often ignored. When a disk dataset is erased, the data by default continues to exist on the disk drive. It is called "residual data" and can easily be read by the next user to allocate a dataset on that part of the disk drive. .) IBM points out in the Security Server RACF System Programmer's Guide Version 2 Release 1 that "***This type of attack requires no exotic tools or insider knowledge and can be done quite easily using JCL and an IBM-provided utility such as IEBGENER.***"

EOS is a RACF option that obliterates data when a disk dataset is erased, so that the data can't be read by the next user of that part of this disk drive. (EOS works only for disk residual data, not for tape, as described in the previous section.)

For years system programmers have objected to using this feature because at one time it had significant performance problems. These problems have since been fixed. (Cheryl Watson and Frank Kyne have shown with hard measurements that EOS is much, much faster with z/OS 2.1 than z/OS 1.13. See details in the last three slides of handout from a NewEra hosted webinar www.stuhenderson.com/Handouts/DontKnow.pdf . They comment that the measurements show such stunning performance improvements that any installation not using EOS should re-visit the issue, once you get to z/OS 2.1. If your audit of mainframe security software hasn't addressed this yet, here is a fine reason to do so.

Understanding the risk of not implementing EOS leads us to an important audit question: who decides and how whether to use it? For some people, the immediate answer is "the RACF administrator" or "the system programmer". But neither of these completely understands the business risks, nor the relevant laws and regulations. The decision of when to implement EOS needs to include input from the relevant business or operational unit, as well as from the Legal or Compliance department.

For effective audits, you might want to evaluate not only whether EOS is used, but how the decision is made, and whether it results in a meaningful risk assessment. Federal agencies are required to have a formal risk assessment for each application. This is useful in the commercial world too. Each application's risk assessment can document which datasets are considered sensitive, who made the decision, and how.

Which do you think is a more meaningful audit comment: "***Erase-On-Scratch is not active and we think it should be. We recommend that it be made active.***" OR

"***The organization has not evaluated the risk associated with unauthorized copying of sensitive data left on disk drives after datasets are erased. Because of this, a feature in the security software which could reduce this risk (a feature called "Erase-On-Scratch") has not been implemented. We recommend that formal risk assessments of this risk be made for each application's datasets, with input from the application owner, Legal, Compliance, system programming, and security administration. We recommend further that when this risk is identified, features such as Erase-On-Scratch be implemented as needed to provide appropriate protection.***"

PASSWORDS

Passwords are a classic case of “There is no single proper setting, you need to understand the big picture and balance the risk.” The starting point in this understanding is to determine who can read the RACF database and any of its copies. This is because anyone who can read the RACF database can run a password cracker program against it to learn all the userids and their passwords. (This was illustrated recently by the 2012 hack of a northern European RACF data center. To learn more, please see the handout from the NewEra-hosted webinar describing the hack at: http://www.stuhenderson.com/StuTop12f.pdf.)

Cracker programs learn passwords by brute force, even if they are encrypted. Making the passwords longer or changing the encryption algorithm doesn’t stop cracker programs, but only adds to how they take to work.) If anyone can read the RACF database, it is easy to describe the resulting risk, leading up to an audit finding.

Assuming no one can read the RACF database, then there are several factors to consider, each of which can reduce the risk of someone guessing someone else’s password, and then being able to impersonate that user. All these password parameters are discussed in the body of this book.

Please note that if an installation decides to use mixed case passwords and/or password phrases, it must first verify that all the programs with sign-on screens have been updated and tested to support this.

Note also that in RACF the word “alphanumeric” means that you must have at least one number and at least one letter, without specifying where in the password they fall. This makes it much harder for a hacker to guess.

Note also the recently added feature Password Minimum Days. If set to zero, this has no effect. If set to one, it prevents a user from changing his own password more than once in a given day.

To give you a start in assessing password option risks, consider whether the following collection of settings would provide sufficient protection against password guessing: minimum length of six to eight alphanumeric characters, revoke userids after three consecutive invalid passwords, do not allow users to change their own passwords more than once a day, require passwords to be changed every 30 days, monitor trends and patterns in the number of invalid passwords and the number of password resets.

An organization can adjust these values as it sees fit, but no one of them by itself can be considered sufficient.

As a side note, has your audit process compared the average number of days it would take a password cracker program to crack most of the passwords to the password change interval? Why would this be relevant?

SECLABELs and MLS

Security labels are an extra layer of security provided by RACF, one which gets checked very early in the access-granting decision process. They are often associated with military and intelligence operations, but are coming to be seen as useful in other venues. The SETR LIST output has many options whose names begin MLS (for Multi-Level Security, which is the security concept which makes use of security labels.) It is not uncommon for RACF installations to ignore SECLABELs. (“Everything has a UACC of NONE and we only permit access that has been authorized and re-certified, so why would we want that?”)

One reason is that they can now add practical controls for DB2 by associating a SECLABEL with a column in a DB2 table. Other uses may occur to you when you consider the need for a separate or more rigorous filtering of access permissions.

Implementing SECLABELs can have surprising side effects, since they can be propagated, and can affect terminals and other resources. One user learned about this the hard way. (Please see http://www.stuhenderson.com/RUGNEW85.pdf for more details.) You will want to read the IBM manual "Planning for MLS and the Common Criteria" cover- to-cover before recommending or implementing SECLABELs or any MLS options.

The moral here for audits is that some options can have practical benefits not mentioned in the checklists.

They can also have impractical and unexpected side effects and costs. This emphasizes the need to understand not only the risks, but also the costs of any audit recommendations.

BASELINE DOCUMENTS AND CONTROL ARCHITECTURE

To step back and look at the bigger picture of how SETROPTS is set, imagine two scenarios. In the first, the RACF administrator decides how to set the options, with input and occasional vetoes from the system programmers. There is no baseline, or documented standard of how the options are to be set in this installation, since the administrator knows it all in his head, and it's what works.

In the second scenario, the security policy assigns responsibility for different parts of the option setting process. Application owners, system programmers, the Legal Department, RACF administrators and others all take part in deciding how the RACF options are to be set. The result is described in a baseline document that specifies how each option is to be set in this installation. Auditors use the baseline as a standard against which to evaluate actual settings. Auditors can evaluate the settings in the baselines to determine if they provide sufficient protection against identified risks.

The first scenario could possibly result in an audit finding that there is no standard (baseline) to audit against, suggesting that the organization itself doesn't understand the associated risks. Perhaps worse, it could result in the auditor using a "one size fits all" checklist as the standard against which to evaluate the actual settings.

The second scenario is easier to audit because there is a written standard to audit against. (RACF administrators will note that auditors are less likely to audit you against some checklist if you have provided your organization's SETROPTS baseline as the standard to be used in your organization.)

It is easier to audit, and to be audited, when there is a clear control architecture. This includes:

- Reasonable assessment of related risks
- Input from people who understand the regulations, the business operations, the system effects, and security administration
- Written baselines specifying how the control options are to be set and
- Easy comparison of actual settings to this baseline.

When an auditor tries to audit SETROPTS settings in an organization without such a control architecture, the lack of a control architecture can become the main audit finding. But only if the auditor provides a description of the resulting risk.

Summary

In short, when deciding how to audit SETR options (or how to set them), you will want to go beyond guides and checklists to understand the risks, the possible protections, and additional benefits. That understanding takes extra work, but pays off, in easier administration, better audits, and more effective security.

The approach described here has several advantages over a "check off the items on the checklist" approach to auditing:

- Gets you to understand the organization and be better regarded
- Involves the people who have the knowledge and the authority
- Results in better security and protection for the organization
- Makes auditing a form of consulting

Two useful sets of guidelines will help you audit SETROPOTS more effectively:

- The NIST STIGs (Security Technical Information Guides) for various types of computer, including mainframes with RACF

  https://web.nvd.nist.gov/view/ncp/repository

- Useful guidelines for knowing that your InfoSec is comprehensive (Note especially Publication 800-53):

  http://csrc.nist.gov/publications/PubsSPs.html#800-53.

SETROPTS - An Opinion

By Mark Hahn

Mark Hahn

The SETROPTS command is an excellent Command Line Interface tool for the specification, validation and display of a system's RACF security settings.

Like most TSO commands built and evolving from the late 1970's, it has a wealth of operands and nested sub-operands - all of which may seem cryptic, but become more easily understood when broken into their component parts, as this book does for the reader.

The SETROPTS command has internal controls to ensure that those settings affecting the security of the user environment require the SPECIAL attribute. In a like fashion, those settings related to audit controls can only be accessed (seen or changed) by users with the AUDITOR attribute. Thus the command implements a separation of function.

Not only can SETROPTS be used to set controls, it will list all of the system security settings in a consistently formatted display. Since it was developed in a terminal oriented period, the listing is very line oriented. This can be a bit frustrating in light of today's panel orientation: the information is all there — but the user must issue it repeatedly if the desired output is overlooked when it scrolls by, or the output must be captured to a dataset and then standard browsing tools utilized. So, it is very complete, but with potentially frustrating displaying characteristics, which are addressed by other services interfacing with SETROPTS.

Not only does the command perform the authorized changes and list the environment, but it is also self-documenting: it cuts an SMF record which contains the means to reconstruct an image of the issued command, so that the setting changes can be tracked and this information used by other system routines. For example, another routine may see the generated SMF record and issue an immediate alert: a console operator message, email, or some other communication, to advise when the command has changed a setting, but not when a SETROPTS LIST (to display the settings) has been issued. By the same token, your SMF auditing reports may simply document that the command had been issued at a specific point in time (not an immediate notification). This allows the user installation to determine the actions to take when the command is employed.

In this fashion security administration may become immediately aware of settings that may potentially reduce the security of the user's environment (for example reducing PASSWORD_MINIMUM_CHANGE_INTERVAL to 0 and allowing multiple password changes in the same day); but auditors will simply find a record of the event in their periodic reviews.

Over the years, the user interface to SETROPTS has evolved:

The command began as, and remains, a TSO line command

In 1984 (RACF 1.6) a series of ISPF panels were developed to gather the operands into a logical collection of related operands (which could also help ensure fewer syntax errors). The panel will build and submit the desired SETROPTS command.

There are now sophisticated interfaces to the SETROPTS command which can format and display

SETROPTS information enabling use overtyping to update the controls. The information is collected, formatted and submitted to the SETROPTS command.

So, in conclusion, the SETROPTS command is a valuable tool for both the system security administrator and auditors to review and control your security environment. It provides separation of function, and can be used to provide both immediate and archival notification of changes it has performed. The user has a choice as to how they interact with the command and its services.

SETROPTS has been and continues to be a valuable resource in the security of the user z/OS environment.

Mark Hahn

Mark Hahn is a retired computer security analyst. His career spanned 30 years of IT - most of it in computer security. His final position was a zSecure Suite Level II Tech Support rep at IBM. His career started back in 1986 when he started attending and speaking at computer security conferences. He has worked for various trail blazing firms as CANDLE Corp as their first computer security administrator, SKK as a tech doc writer and product analyst for acf2/MVS and for Consul Risk Management as tech support for Consul products back in 1996. During this time he has also been active in various security groups such as ISACA, SHARE and the Southern California RACF User Group (SCRUG). Mark has been writing and presenting on computer security for 30 years now and enjoys sharing his experiences. He is also an avid photographer, reader and technology buff.