#SHAREorg

The State of Mainframe Security-or lack thereof

Brian Cummings
Tata Consultancy

Mark Wilson
RSM Partners

Monday, March 10, 2014: 11:00 AM-12:00 PM
Session 14758

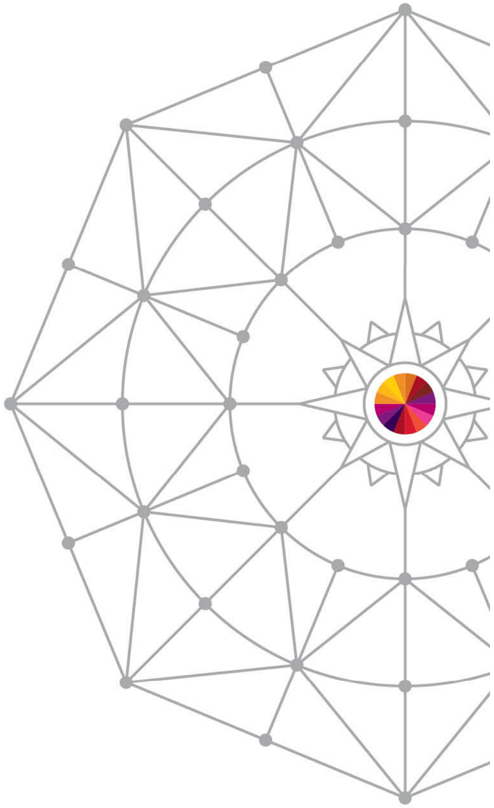
SHARE in Anaheim

Abstract

The speakers will draw on recent experiences, from a combined 60+ years of experience in mainframe security, and from these discussions and projects, to expose real and relevant vulnerabilities in the implementation of mainframe security that create broad exposures to inside and outside intrusion and compromise, negating security effectiveness and SOX and Privacy compliance. The speakers will address common mainframe security myths, and baseline controls and tools that should be implemented.

Agenda

- The simple, low-hanging fruit
- The sublime, technically esoteric
- The scary, the hackable mainframe



Opening Thoughts

1. We have been talking about this for a long time.
2. Yet, we go from place to place and see the same flaws, and sometimes surprising new ones.
3. Yes - The mainframe is the most securable computing platform – few would argue.
4. Equally Yes, for a variety of reasons, systems and security folks can misconfigure the system or poorly implement security leaving the system in some cases entirely vulnerable.
5. And, Yes, the mainframe still carries the heaviest workload of core processes in most large enterprises.
6. And, No, the mainframe is not going away, and we see the pendulum of attention swinging back to it.

Are We Telling the Right People?

- Brian – 2006, 2007: Regulatory Compliance
- Brian – 2009: Sustaining z Security
- Emrich & Valyo – 2009 : Critical z Assessment Findings
- Emrich – Multiple & Here: Top Ten Recurring z Security Audit Findings
- Brian & Mark Hahn - 2012: How secure is your mainframe – Really
- Lennie & Jamie (UK): Recurring z Security Audit Findings
- Hans Schoone - Multiple: Ways to bypass z controls
- Paul Robichaux – Multiple: z/OS Configuration Failures; Mainframe Audit Findings; Gaining Control of SysProgs
- Mark Wilson – Multiple: Mainframe Penetrations

Have you set the rules?

Security Configuration & Access Management frequently devolves to a state of manager approvals. If a request is approved by a manager, the request is processed.

Yet, what governance is there over such manager approvals?

How is consistency achieved across the managerial ranks?

How is good security practice assured?

Technical Security Implementation and Administration Standards

ACF2, RACF, Unix, Windows

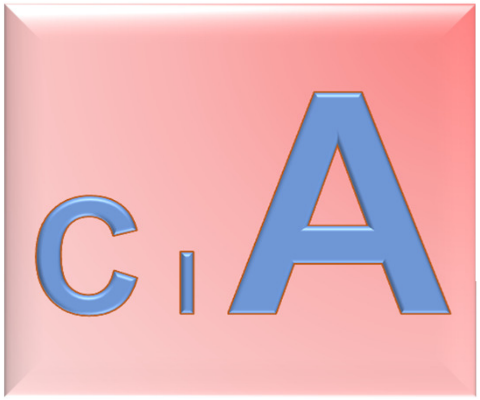
Governance Document to Assure Globally Effective & Consistent Security Practices

a.k.a.
"Adult Supervision"

Per the 2013 Verizon Data Breach Report, 76% of intrusions exploited weak or stolen credentials; 78% of intrusions were rated "low difficulty". Fundamentally, this represents extensive poor security practice.

Security Implementation Standards are essential to achieve consistent and effective security and access management across an enterprise. Simple reliance on manager approval does not achieve good results.

Is your strategy balanced?

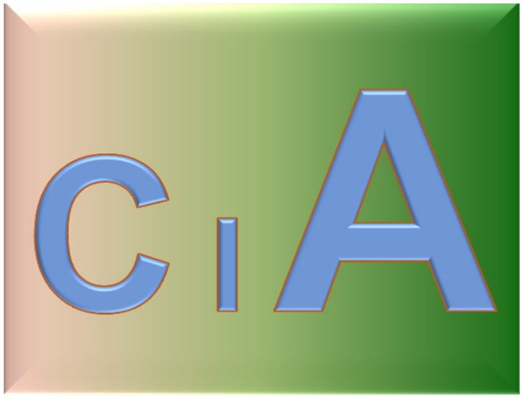


In our experience, entities tend to emphasize Availability over Confidentiality and Integrity in the Security paradigm.

In a worse case, we have seen where all prudence is set aside and “owners” are simply given what they request, without any governing rules.

In these cases, it becomes ever more impossible to even assure Availability, much less to assure and prove compliance and integrity.

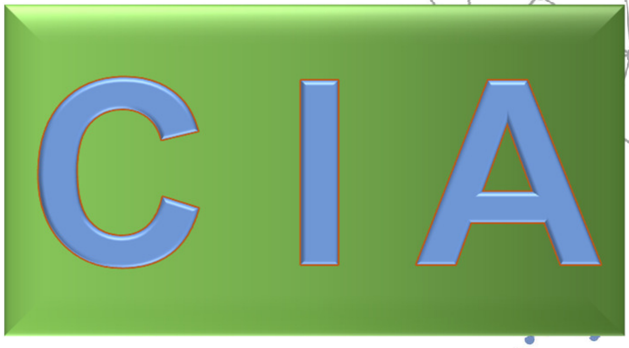
Financial & Privacy Regulation



Privacy regulations drove an increase in focus on Confidentiality, and SOX likewise for Integrity, but still Availability has reigned supreme. We still see severe foundational weakness in infrastructure level security that can undetectably undermine application and business process controls.

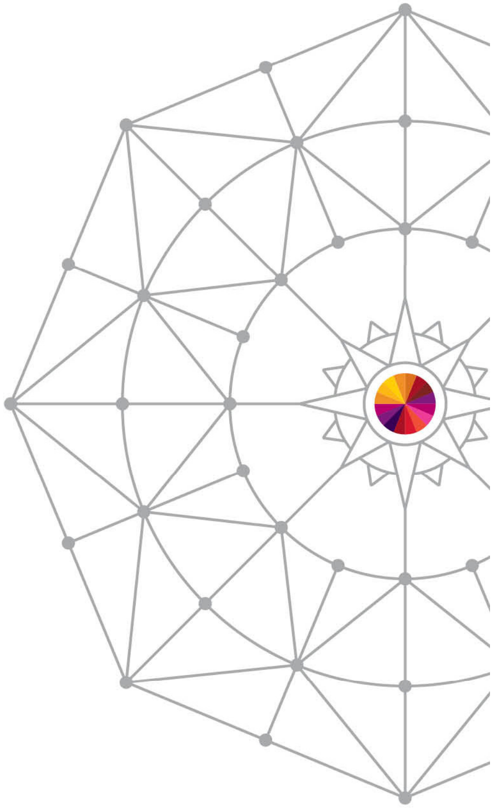
An enterprise needs to understand whether it has C-I-A balance; establish governing rules for security management to define an assure that balance; and align current security controls and practices with those governing rules.

Program Optimization & Maturity



The Simple, Low-Hanging Fruit

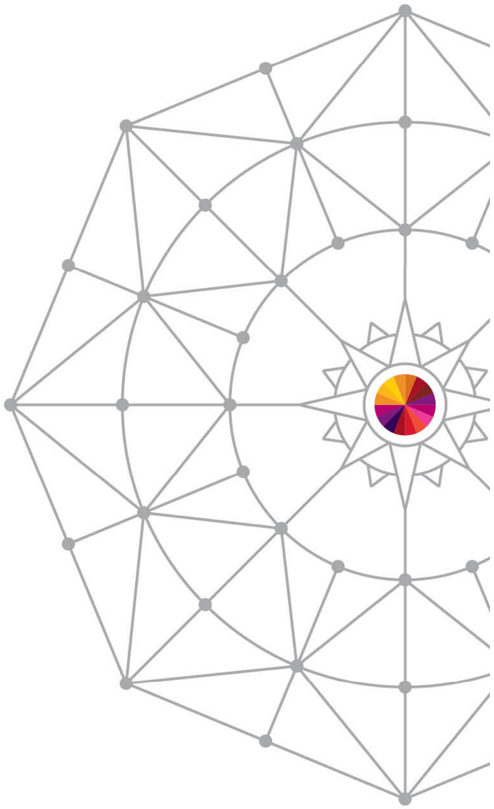
- Identity Management
- Password Management
- Provisioning and De-Provisioning
- Privileged Accounts
- Shared Accounts & Segregation of Duties
- Event Monitoring



The Critical and The Recurring

(in no particular order)

Technical		Governance
IDs with non-expiring passwords		Mainframe is Inherently Secure
Inappropriate USS Super Users		No Security Design
Generally poor USS resource security		No Implementation Standards
UACCs READ or greater		No Continuous Improvement
APF Library Accessibility	Or Shared Account	No garbage cleanup
Highly Privileged Batch Default	Or Shared Account	Security diminished to Admin activity
Highly Privileged STC Default	FAIL or ABORT MODE or PROTECTALL	Lack of skills and training
Security Not Enforced		Audits insufficiently deep and technical
Excessive Account Privileges		Insufficient Monitoring, Alerts, Reporting

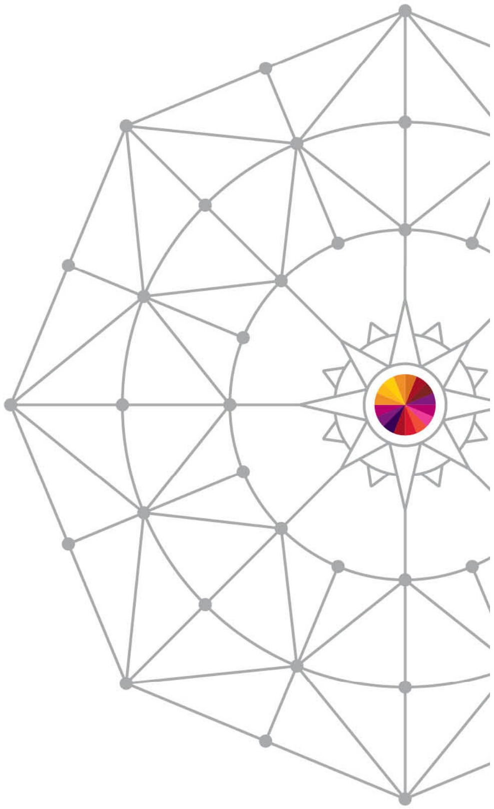


Common Objections

- We Pass Our Audits
  - Ask yourself what your auditors look at.

- We have never been breached
  - How would you know?
  - Just because you have not found them doesn’t mean they are not there?

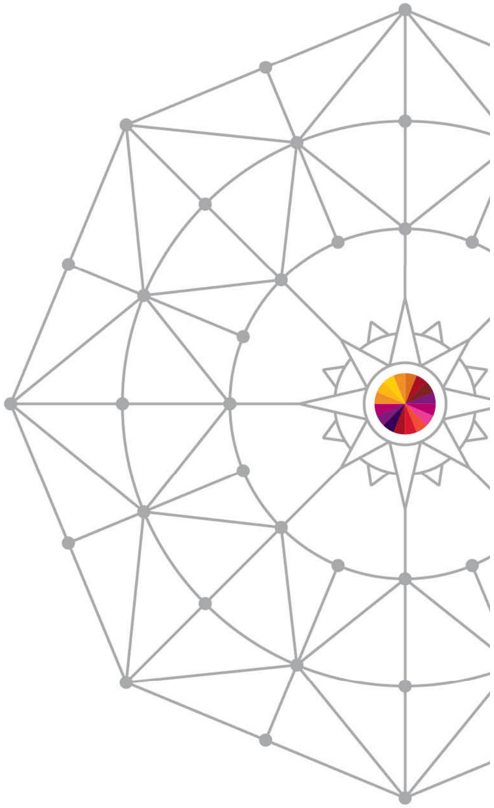
- There is no evidence of a mainframe breach
  - Ummm…check that premise.



The Scary, The Hackable Mainframe

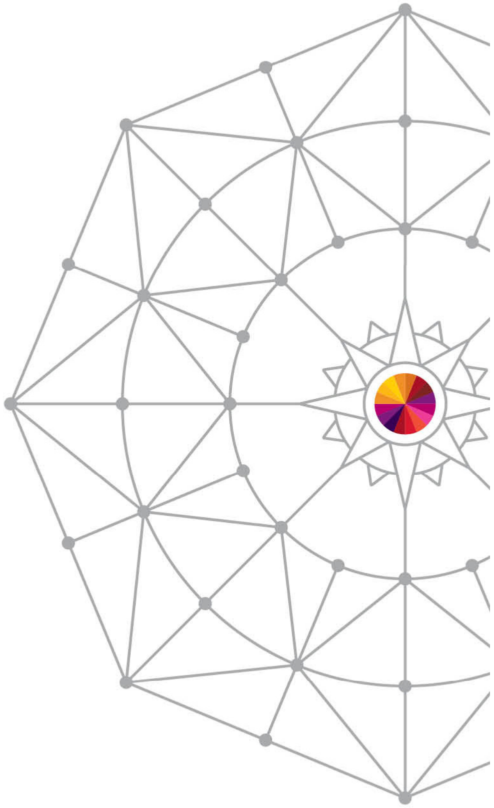
Can a Mainframe be hacked?

- Long running Linkedin discussion started with a very simple question, but a very serious message:
  - ***Is it possible to hack mainframe system?***
    - ***I want to know whether its possible to hack mainframe system. In my Fresher Learning program I heard that mainframe system cannot be hacked, is it true?***
- Who told the Fresher a mainframe could not be hacked?
- How many others think that?



Can a Mainframe be hacked?

- Biggest misconception here is people believe mainframes (zOS and associated subsystems) cannot be hacked!
- I hope everyone in this room knows that’s not true!
- Mainframes do get hacked, but for obvious reasons we rarely hear about them
- The biggest issue is still insider threat, but I have seen an external hack work!
- Mainframes usually mean, "Big data” and that's what the serious hackers want



IBM Mainframe Hacked I

- Swedish Man Charged with Hacking IBM Mainframe & Stealing Money - Apr 16, 2013
- Gottfrid Svartholm Warg was charged with hacking the IBM mainframe of the Swedish Nordea bank, the Swedish public prosecutor said on Tuesday.
- "This is the biggest investigation into data intrusion ever performed in Sweden," said public prosecutor Henrik Olin.
- A large amount of data was taken during the hack, including a large amount of personal data, such as personal identity numbers.

IBM Mainframe Hacked II

- Recent case in the UK
- Senior Applications developer
- Detailed knowledge of the application
- Exploited a known security control
- Defrauded his employer of over £2,000,000 (Sterling)…Just how many $$ is that!

Simple Stuff – RACF Controls

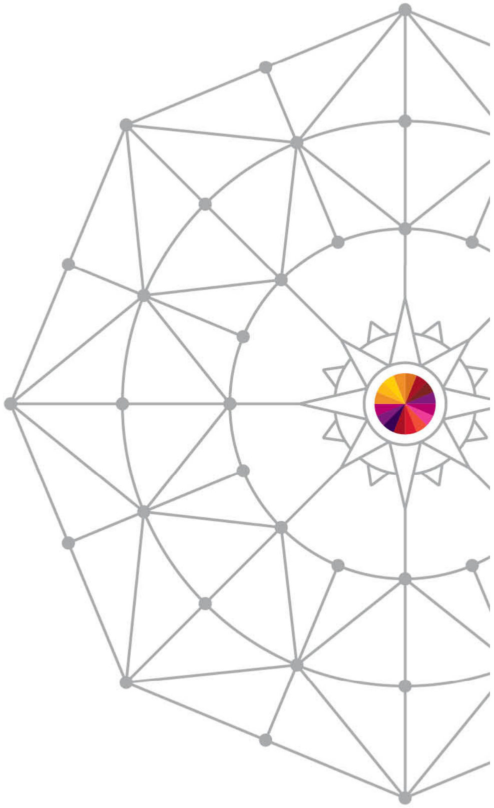
- SURROGAT Class profile
  - FRED.SUBMIT
  - UACC(NONE) with ID(*) ACC(READ) on ACL
- The Userid FRED had the OPERATIONS attribute
- Was able to read, update, delete or define most datasets
- This included a daily unload of the master client database, they were a credit card processing company
- Downloaded this to USB stick and presented this to the CIO
- It showed amongst others things his own credit details and credit rating!

Simple Stuff – RACF Controls

- OPERCMDS Class profile
  - MVS.SET.**
  - MVS.** UACC(NONE) with ID(*) ACC(READ) on ACL
  - But no MVS.SET*.** or MVS.SETPROG.**
  - What's the difference:
    - MVS.SET. Protects the T PROG= command
    - MVS.SETPROG. Protects the SETPROG command
- Was able to
  - Issue the SETPROG command adding my own LOAD library to the list of APF authorised libraries
- I know have control of the system

The Sublime, More technical

- The following slides are not for the faint hearted and assume a certain level of technical understanding
- During our many years of doing this type of work we have seen issues with:
    - z/OS Configuration
        - SVC & PC Routines
        - APF, Linklist & PPT
    - User Identity
        - Passwords
        - Surrogat
        - UID/GID
    - UNIX mechanisms
        - mount and file attributes



What state am I in?

- And I don’t mean California!
- Programs run in two states on a mainframe:
  - Problem State
    - Which is where NORMAL User/Applications Run
  - Supervisor State
    - Where the good stuff is done
    - Use MODESET to switch
    - MODESET is protected
    - You must be Authorised to switch
    - Authorised is protected
      - *Loaded from an APF Authorised Library*
      - *Linkedited with AC=1*
  - Often referred to as a MAGIC SVC

Poorly Coded SVC’s

- Recently performed a test for a large multi national organisation…..
- Basic RACF controls were very good
- However, we found several poorly coded SVC’s, that would allow a user to switch to supervisor state in an uncontrolled manner!

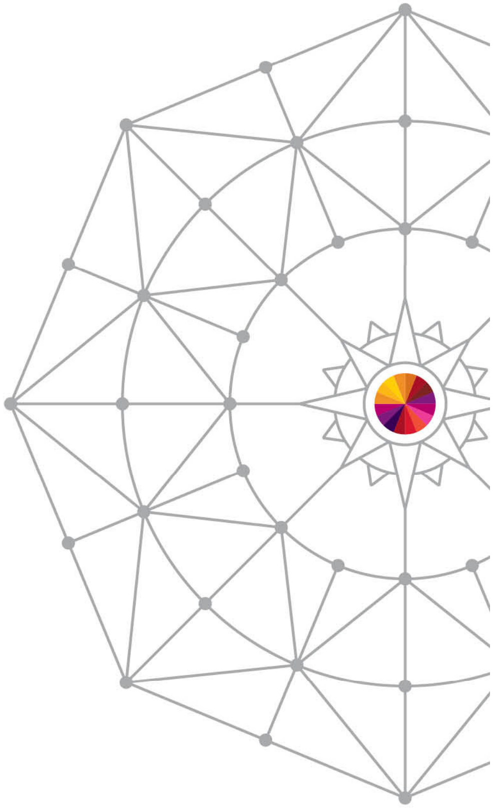
Poorly Coded SVC’s

- SVC 2xx allow a user to gain control in APF-Authorised Status by issuing the SVC with the character string “AUTH” in Register 1

- So a little piece of code in the wrong hands:

```
ICM   R1,15,=C'AUTH’
SVC   211                    AUTHORIZE ME
MODESET KEY=ZERO             SWITCH TO KEY 0
```

- Comment from one of the customer techies: “That was a good spot…how did you do it”



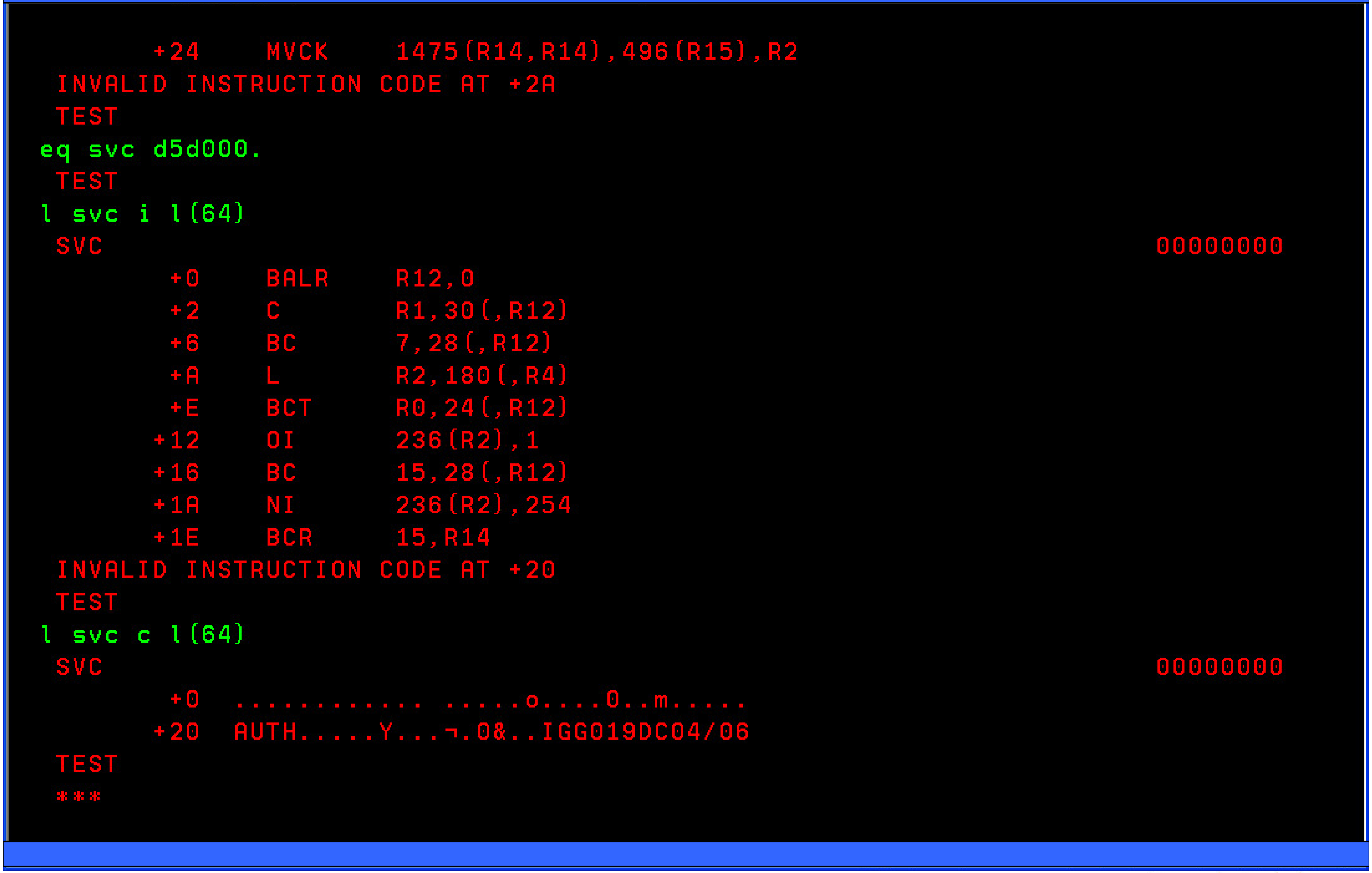
Poorly Coded SVC's

- Installed TASID which displays the SVCTABLE
- Noted the offsets for each installation defined SVC
- Used the TSO TEST command to list the beginning of each SVC
- At offset x'02' the SVC compares the contents of Register 1 to the character string AUTH
- If it matches, then it loads Register 2 with the contents of Register 4 + x'B4'
- On entry, Register 4 contains the address of the Task Control Block (TCB)
- Offset x'B4' into the TCB is the address of the Job Step Control Block (JSCB)

Poorly Coded SVC’s

- Then, at offset x’12’, the SVC issues an Or-Immediate instruction (OI) that turns on the x’01’ bit at offset x’EC’ into the JSCB
- This bit is defined by IBM as:

  "X'01'" - THE STEP REPRESENTED BY THIS JSCB IS AUTHORIZED TO ISSUE THE MODESET MACRO INSTRUCTION. ALTHOUGH THIS BIT HAS BEEN DESIGNATED PSPI, IBM RECOMMENDS THAT VERY CAREFUL DESIGN CONSIDERATION BE GIVEN TO IT'S USE.
- Once this authorised attribute (bit) is turned on, the executing program can issue the MODESET KEY=ZERO macro and z/OS will place it into Key 0
- You now have CONTROL with a Capital K!



```
       +24    MVCK    1475(R14,R14),496(R15),R2
 INVALID INSTRUCTION CODE AT +2A
 TEST
eq svc d5d000.
 TEST
l svc i l(64)
 SVC                                                                 00000000
        +0    BALR    R12,0
        +2    C       R1,30(,R12)
        +6    BC      7,28(,R12)
        +A    L       R2,180(,R4)
        +E    BCT     R0,24(,R12)
       +12    OI      236(R2),1
       +16    BC      15,28(,R12)
       +1A    NI      236(R2),254
       +1E    BCR     15,R14
 INVALID INSTRUCTION CODE AT +20
 TEST
l svc c l(64)
 SVC                                                                 00000000
        +0  ............ .....o....0..m.....
       +20  AUTH.....Y...¬.0&..IGG019DC04/06
 TEST
 ***
```

Are people interested….

- In mainframe security?
- Well why not Google the phrase “Soldier of Fortran”
- http://mainframed767.tumblr.com/
- And have a good read of what people are discussing about mainframe security

Executing Commands on z/OS Through FTP

By Philip “Soldier of Fortran” Young

Presented July 2013 at Black Hat USA

“…z/OS…like any platform, through lack of strong security controls, lack of understanding of the underlying operating system and outward threats, the system can be compromised.”

“Historically the security community has done a poor job of evaluating and pushing the limits of z/OS security. Be it the foreign architecture, the outdated thinking that these platforms are no longer in use or, most likely, the lack of access to the operating system, z/OS has been able to fly under the radar of security professionals.”

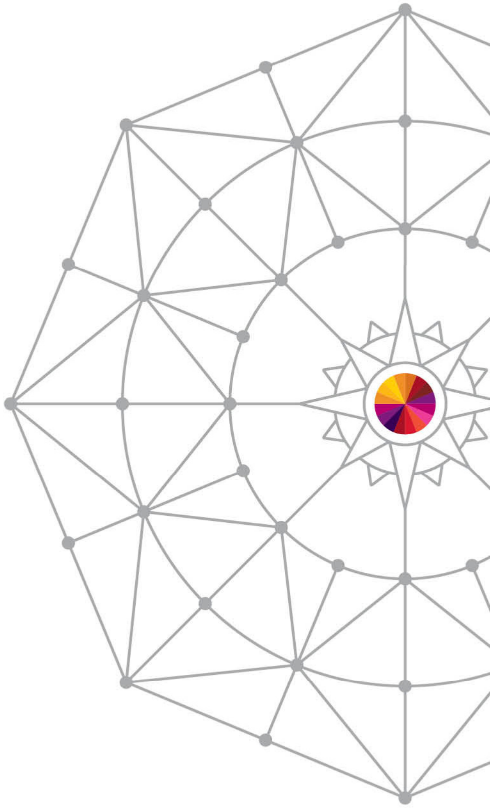
“Similar to the UNIX based FTP daemons, the z/OS daemon provides users with access to their files stored on the mainframe.”

“Without robust security testing programs, features and bugs may exist that expose these systems to undue risks.”

Adobe Acrobat Document

The “Epic Fail” of z Security

- System configuration and system security implementation failures result in vulnerabilities that impair security, privacy, and compliance. None of these can be assured.
- If you emphasize the “A” in C-I-A, you facilitate the compromise of security and ultimately of data at a level that cannot be detected at the time of the event by application and business level controls. Nor could the actions be determined through post-event investigation.
- Independent Financial Auditors beware if you are placing your bets on application and business process controls.



Where to go from here?

- Self Assessment (check some of these issues)
- Set standards for z System Config & Security Implementation
- Professional Assessment (not an audit)
- Lock down privileged users and bypass privileges
- Active Monitoring (few do enough)
- Mitigation (could be a challenge)
- Testing and Validation
- Regular Reassessment (not an audit)
- Training (z/OS and ESM)
- Software Solutions:
  - IBM zWatch (Keeping an eye on everything)
  - New Era Image Sentry (Keeping an eye on the SysProg)
  - zSecure (automate security & compliance checking)

Is everyone working? Together?

The stakes have increased, and where various technology, risk, audit, compliance, and security teams are not all helping and on the same page at an enterprise, we repeatedly see the counter-productive outcomes: Oversights; Errors; Competition; working at cross purposes; lack of cross-team project support. The impact is clear: Wasted time, wasted money, and increased security vulnerability.

If your entire technology, risk, compliance, security, and audit teams are not all pulling in the same direction, you will not get where you need to be as fast as you need to be there.

Thank You!