Can CICS Be Hacked? Are Yesterday's Practices Today's Exposure?

Leigh Compton
CICS Technical Specialist
IBM zGrowth Team

Abstract

Once upon a time, it was easy to secure a CICS system. Users accessed CICS from terminals directly connected to the mainframe or on dedicated leased circuits. But consider today's CICS. Many users access CICS across the Internet. Yeah, the global backbone network now brings traffic to our CICS systems. Have you updated the security for your z/OS system and for CICS? This session will survey potential entry points for an unauthorized user and the security mechanisms to close that door. The speaker will also highlight security enhancements recently delivered for CICS.

Can CICS be hacked?

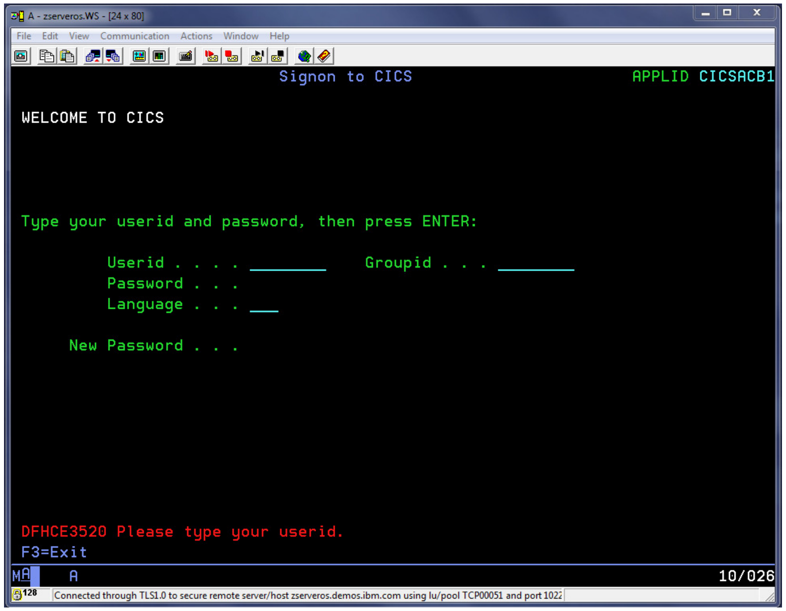
Maybe

What are we talking about?

- What do we mean by “hacking CICS”?
  - Let’s focus on access to CICS
    - Unauthorized
    - Impersonator
- What’s the point of entry to CICS?
  - Terminals/Emulators
  - Browsers
  - Service requests (HTTP)
  - Connectors (CICS TG, WOLA, MQ)
- What protections are in place?
  - Transport security
  - Encryption?

Traditional CICS security

- Sign on (i.e. Authentication)
  - User ID plus password (or pass phrase)
  - Terminal oriented
- Authorization checks
  - z/OS Security Manager
    - RACF, ACF2, or Top Secret
  - Transaction security
  - Resource security
  - Command security

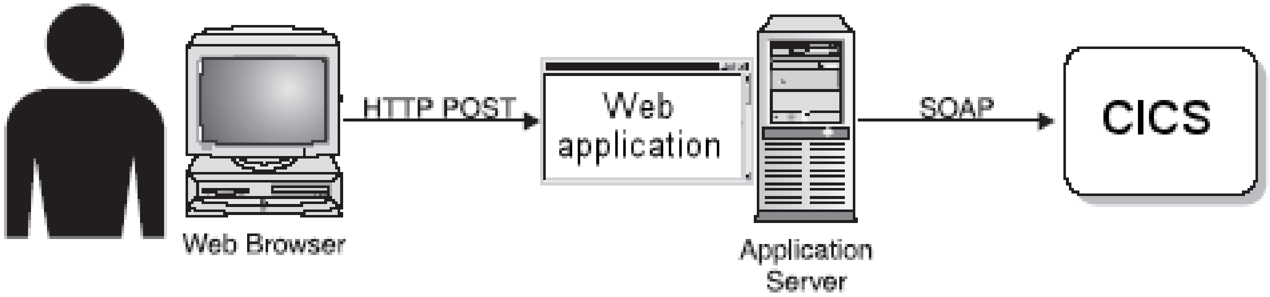

A - zserveros.WS - [24 x 80]
File Edit View Communication Actions Window Help

```
                              Signon to CICS                        APPLID CICSACB1

WELCOME TO CICS




Type your userid and password, then press ENTER:

         Userid . . . . ________     Groupid . . . ________
         Password . . .
         Language . . . ___

     New Password . . .









DFHCE3520 Please type your userid.
F3=Exit
MA     A                                                                     10/026
```

128 Connected through TLS1.0 to secure remote server/host zserveros.demos.ibm.com using lu/pool TCP00051 and port 1022

Security for modern CICS applications

- Non-terminal access
  - No “sign on” as such
  - Requires alternate means of authentication
    - EXEC CICS VERIFY PASSWORD
    - X.509 Certificate
- Intermediate systems



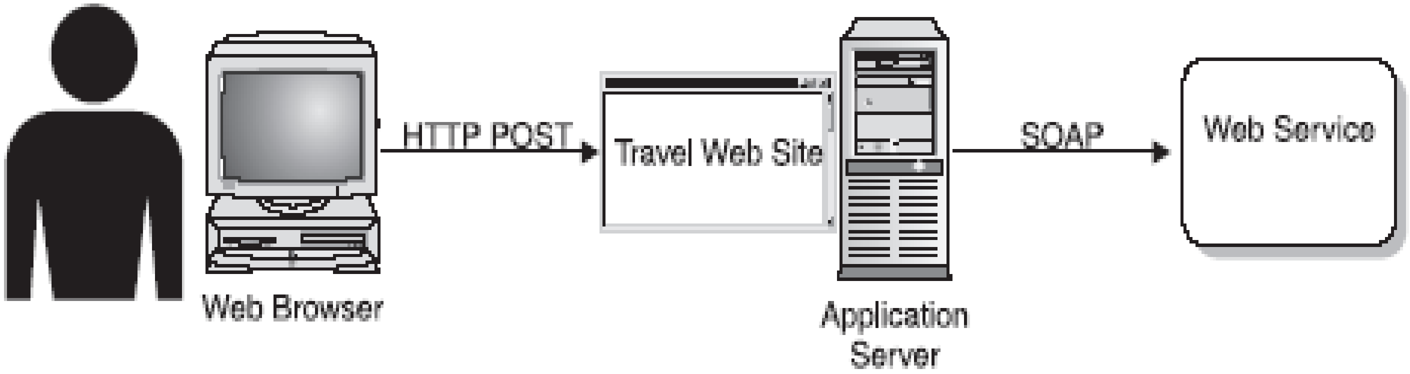
Authentication

- Real or genuine, from Greek: αυθεντικός = 'authentes' = author
- Establishing or confirming something (or someone) as authentic
  - Claims made by or about the thing are true
  - Authenticating a person often consists of verifying their identity.
- In computer security:
  - The process of attempting to verify the digital identity of the sender
    - Such as a request to log in
- Sender being authenticated may be:
  - A person using a computer
  - A computer itself
  - A computer program.

-- Wikipedia -- http://en.wikipedia.org/wiki/Authentication

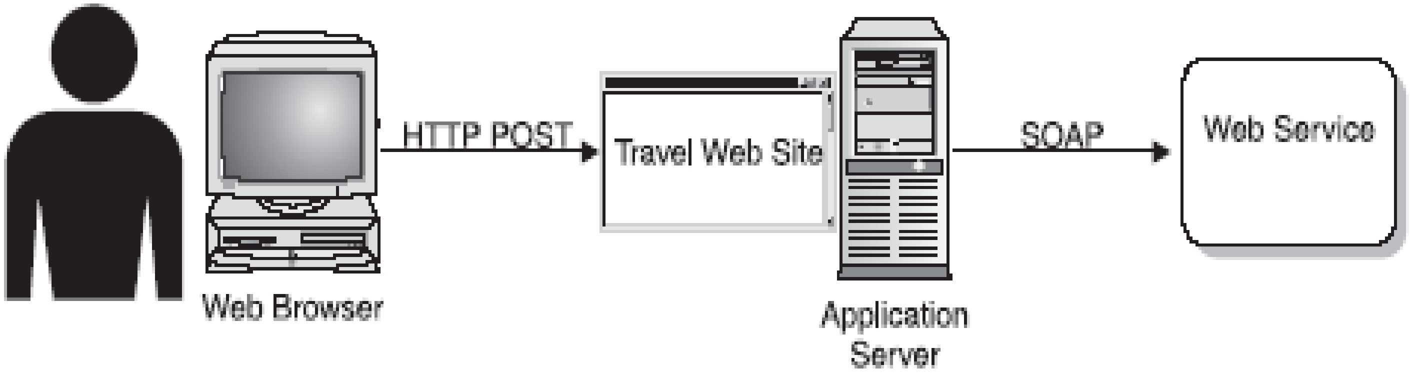
Identity Propagation

- Supports a downstream server in accepting the client identity that is established on an upstream server, with credentials that allow for authentication.



Identity Assertion

- Supports a downstream server in accepting the client identity that is established on an upstream server, without having to authenticate again. The downstream server trusts the upstream server.



Authorization

- A part of the operating system that protects computer resources by only allowing those resources to be used by resource consumers that have been granted authority to use them.
- Resources include individual files or items data, computer programs, computer devices and functionality provided by computer applications.
- Examples of consumers are computer users, computer programs and other devices on the computer.

-- Wikipedia -- http://en.wikipedia.org/wiki/Authorization

Confidentiality

- Assures that information in storage and in-transit are accessible only for reading by authorized parties.
- Encryption is used to assure message confidentiality.
  - Encryption is the process of scrambling data so as to render it unreadable to all but the holder of the correct decryption key

Vulnerabilities of terminal access

- Almost all terminal access to CICS is TN3270
  - TCP/IP communication
  - Often over the Internet
  - Are you using SSL/TLS encryption on connection?
- Terminal emulator software running in workstation
  - Dark fields?
  - Scripting routines?
  - Can user modify software settings?

Vulnerabilities of connectors

- HTTP Basic Authentication
  - User ID and password encoded with base64
  - Easily extracted from transmission and decoded
  - Should only be used over encrypted sessions (https://)
- WS-Security standards for identity
  - Be aware of those standards which pass credentials in the clear
- CICS Transaction Gateway
  - Credentials flow as part of request
  - SSL only available over IPIC connections
  - Identity assertion allowed on secure IPIC connections

Securing the transport

- Secure transport can provide
  - Encryption
  - Client authentication
  - Mutual authentication
- CICS supports Secure HTTP messages
  - `https://hostname:port/ ...`
  - Using SSL or TLS
- z/OS Communication Server provides secure telnet and tn3270

Encryption on z/OS

- Software routines
  - System SSL
  - AT-TLS
  - ICSF – Integrated Cryptographic Service Facility
- Hardware assist

Cryptographic Hardware on z Systems

- CPACF -- Central Processor Assist for Cryptographic Functions
  - Included on every z System Processor
  - Activated by microcode
  - Implements SHA-x and AES algorithms
  - Optional DES/TDES enablement (feature 3863)
- Crypto Express Feature
  - Available on z System Processors
  - Separately priced
  - Cryptographic engines
    - Coprocessor
    - Accelerator

What else?

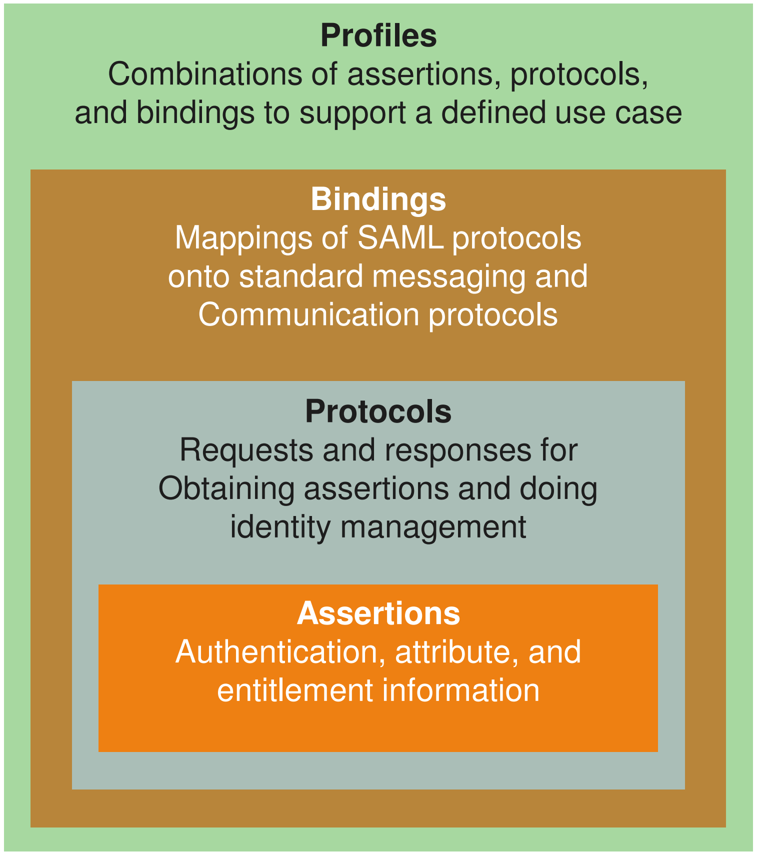
- User authorized to access CICS
- What is user permitted to do?
- Transaction security
  - Which transID is user authorized to execute?
- Resource security
  - Which resources (files, programs, queues, etc.) may the user access?
  - At which level? (read, update ?)
- Command security
  - Which system programming commands may the user execute?
  - INQUIRE only? SET or PERFORM?

What's New in CICS Security

- Web Services Security (WS-Security)
  - Username Token – V.3.1
  - X509 Token – V3.1
  - SAML Token – V5.2 (Feature Pack for V4.2 & V5.1)
  - Kerberos Token – V5.2
  - Message Encryption – V3.1
  - Digital Signatures – V3.1
  - WS-Trust – V3.2
- Enhanced Cryptographic Support – V5.1
  - TLS 1.1 & 1.2
  - Extended Cipher Suites
- Disable SSL V3.0 – via PTF

SAML – Security Assertion Markup Language

- OASIS open standard
- “XML based framework for describing and exchanging security information between on-line business partners.”
- SAML dates from 2001; most recent update from 2005
- Used for:
  - Web SSO
  - Attribute-based authorization
  - Web service security



Profiles
Combinations of assertions, protocols, and bindings to support a defined use case

Bindings
Mappings of SAML protocols onto standard messaging and Communication protocols

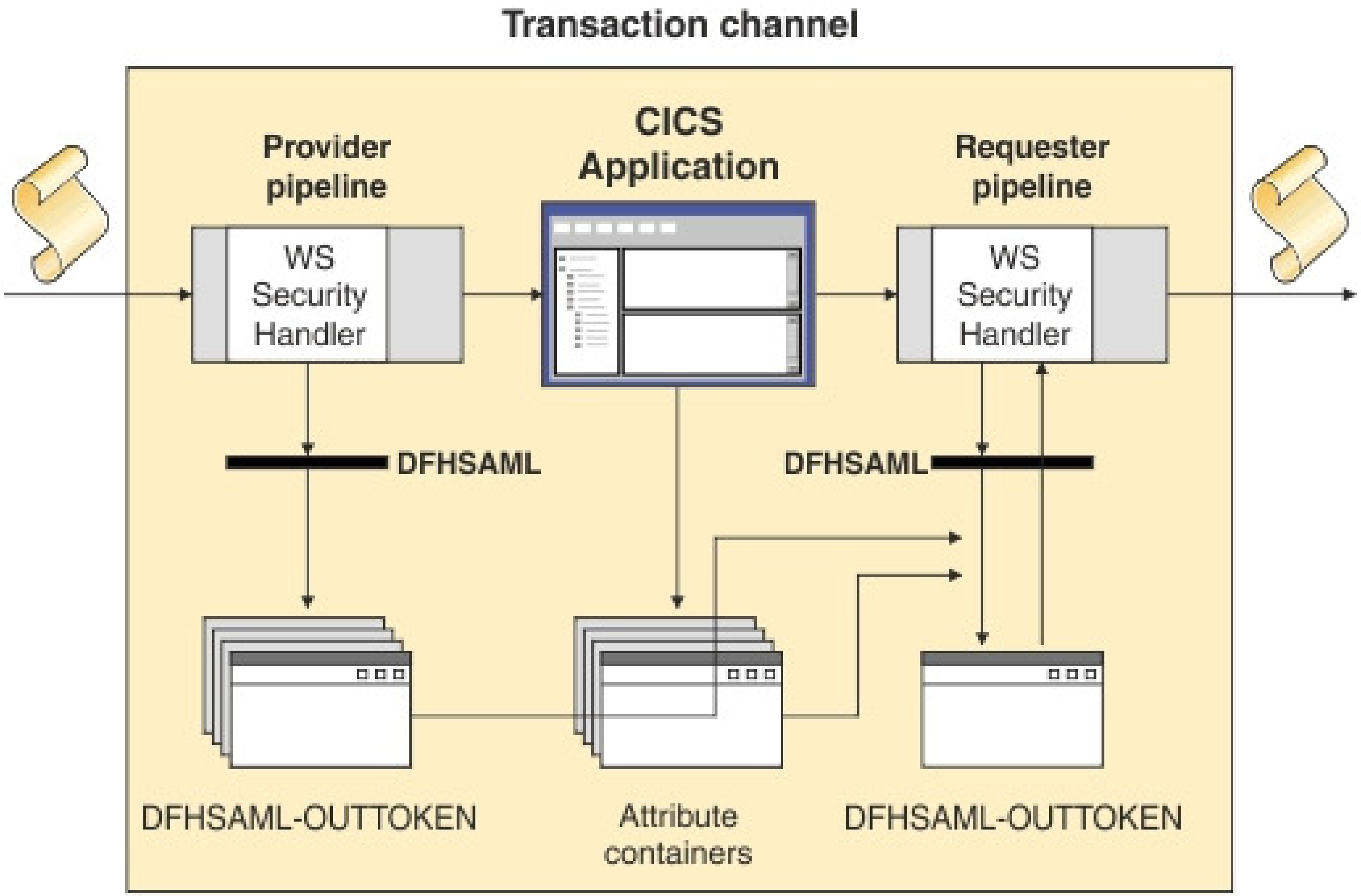
Protocols
Requests and responses for Obtaining assertions and doing identity management

Assertions
Authentication, attribute, and entitlement information

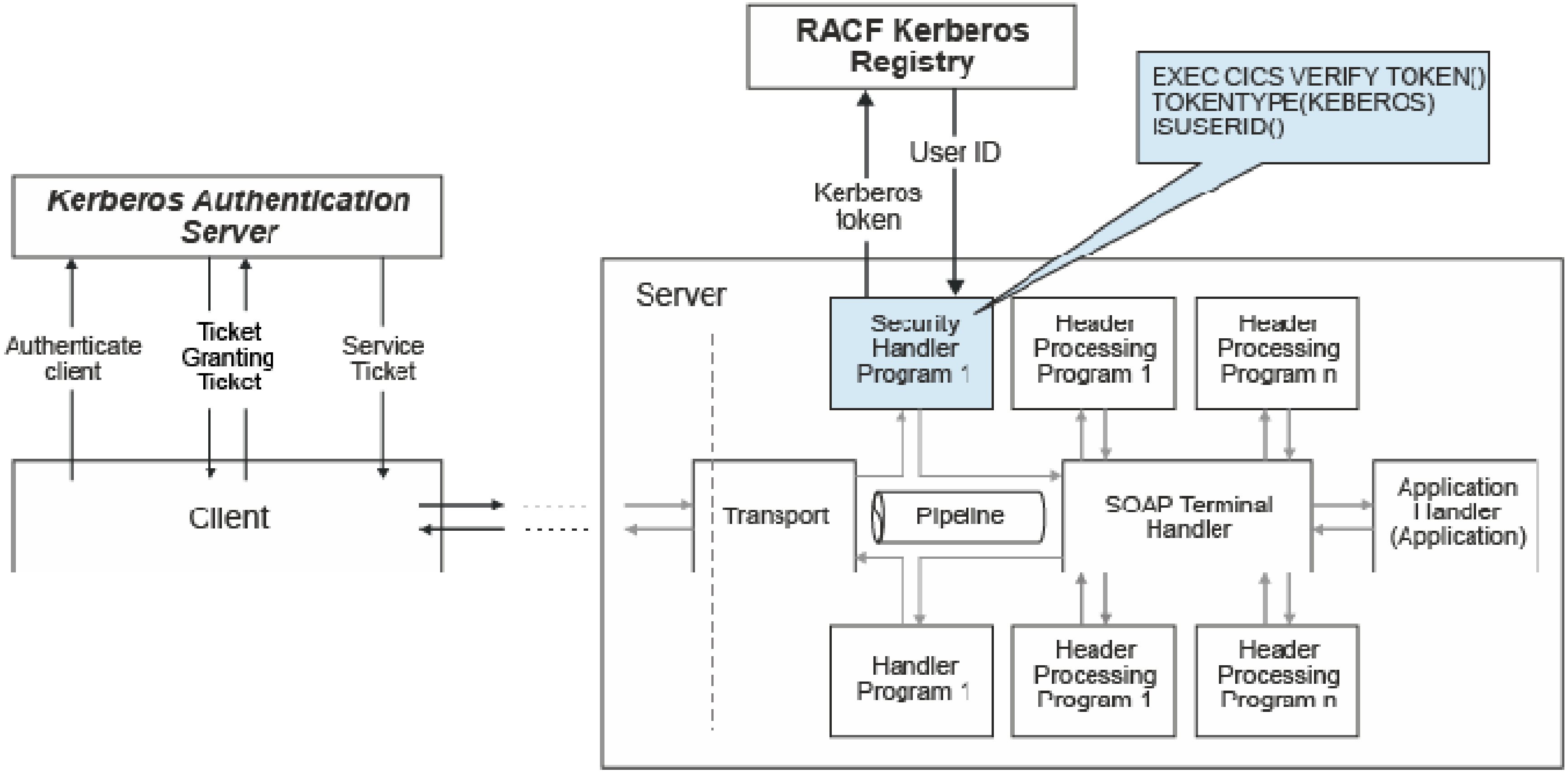
CICS Support for SAML

- SAMLCore1.1 and SAMLCore2.0
  - No support for protocols
- WS-Security authentication
  - Token validation
  - Extraction of SAML parts for inbound messages
  - Addition of SAML token to SOAP request
  - Augmentation of SAML token before it is added to outbound message
- API
  - Linkable interface: DFHSAML
  - Channel and containers
  - Create tokens
  - Validate tokens
  - Extract SAML parts
  - Augment SAML assertions

SAML processing in CICS



Kerberos and CICS



Verify Token

```
>>-VERIFY TOKEN(data-area)--TOKENLEN(data-value)---------------->

>--+---------------------------+-------------------------------->
   '-+-TOKENTYPE--(--cvda--)-+-'
     '-KERBEROS--------------'

                                  .-BIT------------------.
>--+---------------------------+--+----------------------+------>
   '-ISUSERID--(--data-area--)-'  +-DATATYPE--(--cvda--)-+
                                  '-BASE64---------------'

>--+----------------------+--+--------------------+------------><
   '-ESMREASON(data-area)-'  '-ESMRESP(data-area)-'
```

Extended support for cryptographic standards

- Support for TLS v1.1 and v1.2
  - APAR PM97207 for CICS TS v5.1
  - Assures compliance with NIST SP800-131A
  - Adds cipher suites from FIPS 140-2
  - Choose FIPS or non-FIPS mode at CICS start-up
- NIST SP800-131A
  - Requirement for US agencies to transition to stronger cryptographic algorithms and longer keys
- FIPS 140-2
  - Security requirements for cryptographic modules

Defining cipher suites

- CIPHERS – a parameter on resource defintions
  - TCPIPSERVICE
  - URIMAP
  - IPCONN
- Specified as
  - String of hexadecimal 2-character values
    - 35363738392F303132330A1613100D
  - CIPHERs XML file
    - allvalidciphers.xml
    - fipsciphers.xml
    - strongciphers.xml

SSL V3.0 Vulnerabilities

- SSL V3.0 is no longer secure
- Security experts recommend disabling SSL V3.0 in clients and servers
- Ability to exclude SSL V3.0
  - APAR changes default; disallowing use of SSL V3.0
    - PI28039 for V5
    - PI27936 for V4
    - PI28038 for V3
  - New SIT parameter ENCRYPTION=SSLV3 can be specified to include SSL 3.0 if clients require it
- More info: https://developer.ibm.com/answers/questions/195589/disabling-support-for-ssl-v3-in-cics-ts.html

Conclusions

- Security has many aspect
  - Authentication
  - Authorization
  - Confidentiality
- Security starts with the point of entry
  - Encryption is your best protection

Further Resources

- WS-Security specifications
  - http://www.oasis-open.org/specs/index.php#wssv1.0
- WS-I Basic Security Profile V1.0
  - http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html
- IBM Redbooks
  - Implementing CICS Web Services, SG24-7206
  - Patterns: Extended Enterprise SOA and Web Services, SG24-7135
- WS-Trust
  - February 2005, http://www.ibm.com/developerworks/library/specification/ws-trust/

Interesting reading

- Securing Web Services
  - http://www.techweb.com/wire/security/20020508_security
- Best Practices for Web services: Web services security
  - http://www-128.ibm.com/developerworks/webservices/library/ws-best11/
  - http://www-128.ibm.com/developerworks/webservices/library/ws-best12/
- Web Service Security: Scenarios, Patterns, and Implementation Guidance
  - http://www.gotdotnet.com/codegallery/codegallery.aspx?id=67f659f6-9457-4860-80ff-0535dffed5e6
- Solving the web services identity crisis
  - http://www.looselycoupled.com/stories/2004/crisis-id0622.html
- Mainframe security changes as Web services arrive
  - http://searchwebservices.techtarget.com/tip/0,289483,sid26_gci1202408,00.html?asrc=SS_CLA_301932&psrc=CLT_26

FYI

- Asymmetric key algorithms (Public-key cryptography)
  - RSA
    - encryption algorithm patented by Rivest, Shamir, and Adleman
    - patent expired in 2000
  - DSA
    - Digital Signature Algorithm, U.S. Government standard
  - DES
    - Data Encryption Standard, U.S. Government standard
  - AES
    - Advanced Encryption Standard, U.S. Government standard
- Cryptographic hash functions
  - SHA-1
    - Secure Hash Algorithm, developed by NSA, U.S. Government standard
    - used in many security applications and protocols
      - TLS, SSL, PGP, SSH, S/MIME, and IPSec
  - MD5
    - Message-Digest algorithm 5