Tutorial II
Computer Security Module

Mike Just

School of Informatics
University of Edinburgh

March 2010

Purpose of Tutorial

- More time on some areas of Computer Security
- Allow for more interaction with a smaller class
  - More interaction
  - More time for questions
- Method for tutorial
  - Learn additional concepts
  - Review Practical II

Tutorial II Themes

1. Email Security
2. Web Security

Email Security

Simple message

From: sender@domain.com
To: recipient@domain.com
Subject: Message subject.
Content-Type: text/plain

Message body text ...

With attachments

From: sender@domain.com
To: recipient@domain.com
Subject: Message subject.
Content-Type: multipart/mixed;
boundary="multipart_mixed"

--multipart_mixed
Content-Type: text/plain

Message body text ...

--multipart_mixed
Content-Type: application/octet-stream

attachment_data ...

--multipart_mixed--

Email Security (2)

Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m

rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
0GhIGfHfQbnj756YT64V

- What is the structure of this encoded information?
  - This has been established by the Internet Engineering Task Force (IETF) through their S/MIME Working Group
  - Cryptographic Message Syntax (CMS)
    - http://www.ietf.org/rfc/rfc3852.txt

Email Security (3)

EnvelopedData ::= SEQUENCE {
  version CMSVersion,
  originatorInfo [0] IMPLICIT OriginatorInfo
    OPTIONAL,
  recipientInfos RecipientInfos,
  encryptedContentInfo EncryptedContentInfo,
  unprotectedAttrs [1] IMPLICIT
    UnprotectedAttributes OPTIONAL }

OriginatorInfo ::= SEQUENCE {
  certs [0] IMPLICIT CertificateSet
    OPTIONAL,
  crls [1] IMPLICIT RevocationInfoChoices
    OPTIONAL }

RecipientInfos ::= SET SIZE (1..MAX) OF
  RecipientInfo

EncryptedContentInfo ::= SEQUENCE {
  contentType ContentType,
  contentEncryptionAlgorithm
    ContentEncryptionAlgorithmIdentifier,
  encryptedContent [0] IMPLICIT
    EncryptedContent OPTIONAL }

EncryptedContent ::= OCTET STRING

UnprotectedAttributes ::= SET SIZE (1..MAX)
    OF Attribute

RecipientInfo ::= CHOICE {
  ktri KeyTransRecipientInfo,
  kari [1] KeyAgreeRecipientInfo,
  kekri [2] KEKRecipientInfo,
  pwri [3] PasswordRecipientinfo,
  ori [4] OtherRecipientInfo }

EncryptedKey ::= OCTET STRING

KeyTransRecipientInfo ::= SEQUENCE {
  version CMSVersion, -- always set to 0 or 2
  rid RecipientIdentifier,
  keyEncryptionAlgorithm
    KeyEncryptionAlgorithmIdentifier,
  encryptedKey EncryptedKey }

RecipientIdentifier ::= CHOICE {
  issuerAndSerialNumber
    IssuerAndSerialNumber,
  subjectKeyIdentifier [0]
    SubjectKeyIdentifier }

Email Security (4)

```
KeyAgreeRecipientInfo ::= SEQUENCE {
  version CMSVersion, -- always set to 3
  originator [0] EXPLICIT
    OriginatorIdentifierOrKey,
  ukm [1] EXPLICIT UserKeyingMaterial
    OPTIONAL,
  keyEncryptionAlgorithm
    KeyEncryptionAlgorithmIdentifier,
  recipientEncryptedKeys
    RecipientEncryptedKeys }

OriginatorIdentifierOrKey ::= CHOICE {
  issuerAndSerialNumber
    IssuerAndSerialNumber,
  subjectKeyIdentifier [0] SubjectKeyIdentifier,
  originatorKey [1] OriginatorPublicKey }


OriginatorPublicKey ::= SEQUENCE {
  algorithm AlgorithmIdentifier,
  publicKey BIT STRING }
```

```
RecipientEncryptedKeys ::= SEQUENCE OF
    RecipientEncryptedKey

RecipientEncryptedKey ::= SEQUENCE {
  rid KeyAgreeRecipientIdentifier,
  encryptedKey EncryptedKey }

KeyAgreeRecipientIdentifier ::= CHOICE {
  issuerAndSerialNumber
    IssuerAndSerialNumber,
  rKeyId [0] IMPLICIT RecipientKeyIdentifier }


RecipientKeyIdentifier ::= SEQUENCE {
  subjectKeyIdentifier SubjectKeyIdentifier,
  date GeneralizedTime OPTIONAL,
  other OtherKeyAttribute OPTIONAL }


SubjectKeyIdentifier ::= OCTET STRING
```

Email Security - Certificates

- As for the data, the certificate content is similarly defined
  - This has been established by the Internet Engineering Task Force (IETF) through their PKIX Working Group, and based upon the ISO X.509 standard
  - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
    - http://www.ietf.org/rfc/rfc5280.txt

Email Security – Certificates (2)

```
Certificate ::= SEQUENCE {
   tbsCertificate TBSCertificate,
   signatureAlgorithm AlgorithmIdentifier,
   signatureValue BIT STRING }

TBSCertificate ::= SEQUENCE {
   version [0] EXPLICIT Version DEFAULT
      v1,
   serialNumber CertificateSerialNumber,
   signature AlgorithmIdentifier,
   issuer Name,
   validity Validity,
   subject Name,
   subjectPublicKeyInfo
      SubjectPublicKeyInfo,
   issuerUniqueID [1] IMPLICIT
      UniqueIdentifier OPTIONAL
      -- If present, version MUST be v2 or v3
   subjectUniqueID [2] IMPLICIT
      UniqueIdentifier OPTIONAL
      -- If present, version MUST be v2 or v3
   extensions [3] EXPLICIT Extensions
      OPTIONAL
      -- If present, version MUST be v3
}

Version ::= INTEGER { v1 (0), v2 (1), v3 (2) }

CertificateSerialNumber ::= INTEGER

Validity ::= SEQUENCE {
   notBefore Time,
   notAfter Time }

Time ::= CHOICE {
   utcTime UTCTime,
   generalTime GeneralizedTime }

UniqueIdentifier ::= BIT STRING

SubjectPublicKeyInfo ::= SEQUENCE {
   algorithm AlgorithmIdentifier,
   subjectPublicKey BIT STRING }

Extensions ::= SEQUENCE SIZE (1..MAX)
OF Extension

Extension ::= SEQUENCE {
   extnID OBJECT IDENTIFIER,
   critical BOOLEAN DEFAULT FALSE,
   extnValue OCTET STRING }
```

Email Security – Certificates (3)

```
KeyUsage ::= BIT STRING {
  digitalSignature (0),
  nonRepudiation (1),
  keyEncipherment (2),
  dataEncipherment (3),
  keyAgreement (4),
  keyCertSign (5),
  cRLSign (6),
  encipherOnly (7),
  decipherOnly (8)
}
```

Web Security – Data & Certificates

- For Web Security, the same certificate format (X.509) is used
- But the data data format is different (and defined by SSL/TLS)

Web Security

- Protecting data on the web can be tricky
- And it's often a compromise between security and usability
- But secure programming practices needed to be followed (and can typically be followed without impacting usability)

Web Security – Flickr Examples

- Flickr is a well-known photo sharing site
- It allows you to upload your own photos
  - Mark as private, disable downloading, ...
- Example – Security through obscurity
  - Private url?

```
http://www.flickr.com/photos/48316726@N04/4424882797/?edited=1

http://farm5.static.flickr.com/4047/4424882797_0f1e01c7f0_m.jpg
```

  - Look, but don't save?

```
http://www.flickr.com/photos/48316726@N04/4425654528/

http://farm3.static.flickr.com/2727/4425654528_c732197bc8_m.jpg
```

Web Security – Practical

- Overview
  - Photo sharing for a camera club 'Edinburgh Shutterbugs'
  - Practical II

Web Security – Practical – Part A

Some examples:

- Photo/photo description is edited by non-creator
  - Vul: Permissions set incorrectly, weak passwords, ...
  - Cost: Slander, embarrassment, …
  - Counter: Offer notification of updates, ...
- Denial-of-service against webserver
  - Vul: CPU usage attack, exhaust storage space, …
  - Counter: IDS, Upload & storage limits, ...
- Denial-of-service against account user
  - Counter: IDS
- Non-member stores too many photos on site
  - Cost: Usage not paid for

Web Security – Practical – Parts B&C

B.1 Info Discovery / C.1.1 Fix

- http://localhost:8000/csphotos/image?name=../../../../../../../../etc/passwd
- Why does it work?
  - '/home/alice/csphotos-data/john/../../../../../../../../etc/passwd' becomes '/../../../etc/passwd' or, the password file
- Fixes
  - Validation of 'name' parameter, e.g., only alphanumeric
  - Validation of 'user' parameter also
  - Check database for user-name pair

Web Security – Practical – Parts B&C

B.2 Denial of Service / C.1.2 Fix

- Example
  - http://localhost:8000/csphotos/image?user=...&name=...&**size=100000**
- Why does it work?
  - Takes advantage of sizing in image script
- Fixes
  - Limit value of the size parameter

Web Security – Practical – Parts B&C

B.3 SQL Injection / C.1.3 Fix

- Example
  - http://localhost:8000/csphotos/album=john'%3B%20UPDATE%20passwords%20SET%20password%20=%20'trivial'%20WHERE%20username%20=%20'john
  - album=john; UPDATE passwords SET password = trivial WHERE username = john
- Why does it work?
  - Takes advantage of embedded SQL commands in album script
- Fixes
  - Filtering/escaping characters
  - Don't directly embed user input into SQL

Web Security – Practical – More?

Additional Observations

-