PKIへの攻撃とその対応【対策編】

セコムIS研究所

島岡　政基

なぜ認証局が狙われるのか?

- セキュリティ警告の実効性が高まってきた
  - オレオレ証明書の検証・インストール
  - コード署名のないソフトウェアのインストール
- いまやパブリック証明書なしには何もできない

⇒ 攻撃者の攻撃力は一層加速すると認識するべき

攻撃の手口と対策の概観

- Comodo/DigiNotar
  - とっかかりはフロントエンドサーバの既知の脆弱性
  - 侵入後はDLL解析とか色々やってるけど。。。
- Flame
  - MD5のchosen-prefix collision attack
  - 証明書のASN.1構造のなりすまし
- Vulnerable Repeated Keys
  - 擬似乱数生成器の脆弱性(エントロピー不足)
  - 運用的な問題だが潜在的な脅威につながる

一般的なシステムの脆弱性

暗号技術の脆弱性

暗号技術の脆弱性

要するに上からも下からもやられてる
(しかも段々高度化している)
しかし、いずれも既知の脆弱性であることに違いはない

業界の動向

- CA/Browser Forum
  - Baseline Requirements v1.0(2012/07～)
    - WebTrust for CAの後継規格となることを意図
  - Network Security Controls v1.0(2013/01～)
    - WebTrust for CA, ETSI 101 456, ETSI TS 102 042が対象
    - セキュリティ要件、権限管理、監視、脆弱性管理

- IETF/wpkops(Web PKI operations, non-WG)
  - トラストモデル、失効処理、TLS運用について標準化予定

お家騒動的な一幕も。。。

問題提起

- 証明書拡張やcriticalフラグに関するルール
  - 暗号ミドルウェア or アプリケーション
    - 拡張capabilityに関するネゴシエーション

  ⇒ CommonCryptoAPI的なもの?

- 監査の限界と透明性
  - 監査人以外はOK/NGしかわからない
  - 到達レベルを把握できる透明性が必要ではないか

  ⇒ Security Level Agreement的なもの?

中長期的な対策

- 認証局ベンダ
  - 脆弱性情報交換組織の確立
    - 暗号学者、セキュリティ技術者との密で速やかな情報共有
    - 速やかかつ効力のある対策ガイドライン策定
  - CA/Browser Forumはもうダメ?
- アプリベンダ
  - CommonCryptoAPI的な何かの標準化
    - CredentialLoA
    - CertCapability
    - エラーコード体系 とか
  - IETF/WebPKopsは期待できる?

PKI業界の責任感に期待する

- PKIシステムが適切な脆弱性対策を行っている限り、攻撃は困難
  - 唯一例外はFlameだが、MD5問題も2007年既報
  - 決して急な対策を求められるわけではない
- しかしComodo/DigiNotarに続く事件が続発すれば信頼されなくなる

- 今より証明書が危険になったら、市場は他のセキュリティ技術へ移行するのか?
  - サーバ認証の代替は?コード署名の代替は?

- 現時点でPKIの代替技術はないし、出現しても代替候補となるには絶対的に時間がかかる
  - 技術の標準化、実装の普及、運用基準の普及、トラストリストの移行、etc.

我々が守らなければならないものは何か?
それを失ったらどうなるのか?

Flame関連

時間に余裕があれば。。。

IssuerUniqueIdentifierの仕様

4.1 Basic Certificate Fields

```
<snip>
        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
<snip>
   UniqueIdentifier  ::=  BIT STRING
<snip>
```

RFC 3280までは SHOULD NOT

4.1.2.8. Unique Identifiers

<snip> CAs conforming to this profile MUST NOT generate certificates with unique identifiers. Applications conforming to this profile SHOULD be capable of parsing certificates that include unique identifiers, but there are no processing requirements associated with the unique identifiers.

RFC 5280より

- 上限なし&BIT STRINGなので何でも突っ込める♪
- これ以外に「上限なし&BIT STRING」なフィールドは、以下の2点だけ。
  - subjectUniqueIdentifier
  - subjectPublicKeyInfo ←これはさすがに大丈夫かと。

serialNumberフィールド

- RFC的には最大20 octetsまで利用可能。
- 今回はそもそも10bytesしか使っていない。
- 予測困難性を実現するのに十分な探索空間を確保できていたか?
  - そもそも攻撃者は4回のコリジョンブロックでコリジョンを発見できる計算リソースを持っている。

Eric Rescorla
TLS WG co-chair

Current status of MD5 and SHA-1, SAAG@62nd IETF

Randomize cert serial numbers!!

神田さん資料より

■シリアルナンバーの予測がかなり難しい

- Microsoft LSRA PAが作るシリアルナンバーはミリ秒単位

Feb 23 19:21:36 2010 GMT	14:51:5b:02	00:00	00:00:00:08
Jul 19 13:41:52 2010 GMT	33:f3:59:ca	00:00	00:05:25:e0
Jan 9 20:48:22 2011 GMT	47:67:04:39	00:00	00:0e:a2:e3

ブート後の経過時間(ミリ秒単位)[4バイト]
CAの識別番号 [固定2バイト]
発行番号[4バイト]

4.1 Basic Certificate Fields

```
<snip>
        serialNumber          CertificateSerialNumber,
<snip>
   CertificateSerialNumber  ::=  INTEGER
<snip>
```

4.1.2.2 Serial number

<snip> Certificate users MUST be able to handle serialNumber values up to 20 octets. Conformant CAs MUST NOT use serialNumber values longer than 20 octets.

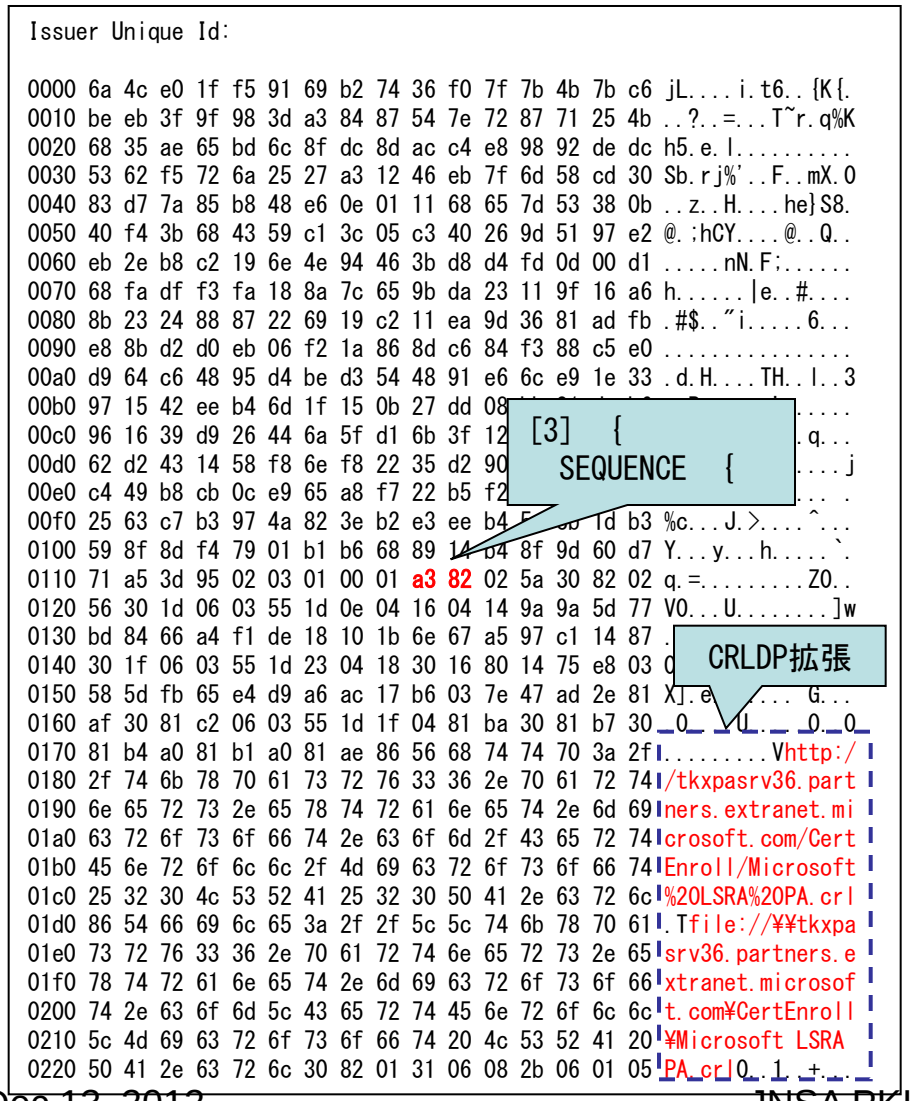
RFC 5280より

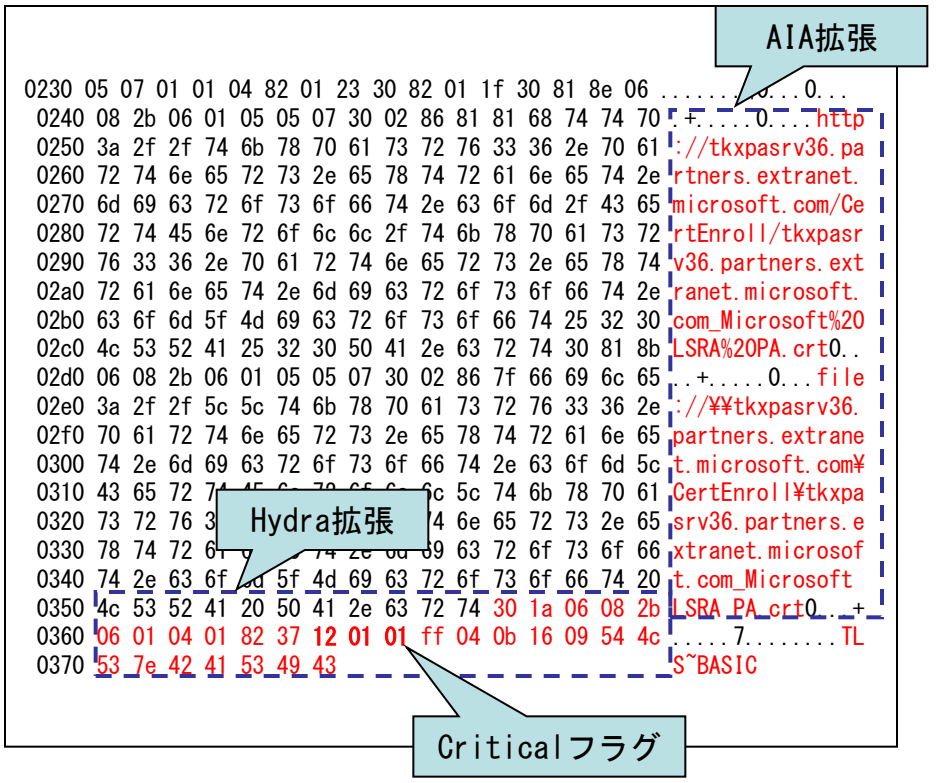
Microsoft Hydra extension

- Terminal Service用のプライベート拡張
  - HydraはTerminal Serviceの開発コード
  - MUST be critical
    # でもXP以前だと無視されるらしい。
  - http://www.oid-info.com/get/1.3.6.1.4.1.311.18
- 偽造証明書にもcriticalフラグがセットされた状態で存在していた。
  - ただしissuerUniqueIdフィールドの中に埋め込まれた形で。

```
  0   26: SEQUENCE {
  2    8:   OBJECT IDENTIFIER '1 3 6 1 4 1 311 18'
 12    1:   BOOLEAN TRUE
 15   11:   OCTET STRING, encapsulates {
 17    9:     IA5String 'TLS~BASIC'
        :     }
        :   }
```

偽造証明書のIssuerUniqueId





Flame malware collision attack explained - TechNet Blogs
http://blogs.technet.com/b/srd/archive/2012/06/06/more-information-about-the-digital-certificates-used-to-sign-the-flame-malware.aspx

偽造証明書のissuerUniqueIDに埋め込まれていた証明書拡張

```
  0 602: [3] {
  4 598:   SEQUENCE {
  8  29:     SEQUENCE {
 10   3:       OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
 15  22:       OCTET STRING, encapsulates {
 17  20:         OCTET STRING
       :           9A 9A 5D 77 BD 84 66 A4 F1 DE 18 10 1B 6E 67 A5
       :           97 C1 14 87
       :         }
       :       }
 39  31:     SEQUENCE {
 41   3:       OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
 46  24:       OCTET STRING, encapsulates {
 48  22:         SEQUENCE {
 50  20:           [0]
       :             75 E8 03 58 5D FB 65 E4 D9 A6 AC 17 B6 03 7E 47
       :             AD 2E 81 AF
       :           }
       :         }
       :       }
 72 194:     SEQUENCE {
 75   3:       OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
 80 186:       OCTET STRING, encapsulates {
 83 183:         SEQUENCE {
 86 180:           SEQUENCE {
 89 177:             [0] {
 92 174:               [0] {
 95  86:                 [6]
       :                   'http://tkxpasrv36.partners.extranet.microsoft.co'
       :                   'm/CertEnroll/Microsoft%20LSRA%20PA.crl'
183  84:                 [6]
       :                   'file://¥¥tkxpasrv36.partners.extranet.microsoft.'
       :                   'com¥CertEnroll¥Microsoft LSRA PA.crl'
       :                 }
       :               }
       :             }
       :           }
       :         }
       :       }
269 305:     SEQUENCE {
273   8:       OBJECT IDENTIFIER authorityInfoAccess (1 3 6 1 5 5 7 1 1)
283 291:       OCTET STRING, encapsulates {
287 287:         SEQUENCE {
291 142:           SEQUENCE {
294   8:             OBJECT IDENTIFIER caIssuers (1 3 6 1 5 5 7 48 2)
304 129:             [6]
       :               'http://tkxpasrv36.partners.extranet.microsoft.co'
       :               'm/CertEnroll/tkxpasrv36.partners.extranet.micros'
       :               'oft.com_Microsoft%20LSRA%20PA.crt'
       :             }
436 139:           SEQUENCE {
439   8:             OBJECT IDENTIFIER caIssuers (1 3 6 1 5 5 7 48 2)
449 127:             [6]
       :               'file://¥¥tkxpasrv36.partners.extranet.microsoft.'
       :               'com¥CertEnroll¥tkxpasrv36.partners.extranet.micr'
       :               'osoft.com_Microsoft LSRA PA.crt'
       :             }
       :           }
       :         }
       :       }
578  26:     SEQUENCE {
580   8:       OBJECT IDENTIFIER '1 3 6 1 4 1 311 18'
590   1:       BOOLEAN TRUE
593  11:       OCTET STRING, encapsulates {
595   9:         IA5String 'TLS~BASIC'
       :         }
       :       }
       :     }
       :   }
```

- authorityKeyID
- cRLDistPoint
- AuthorityInfoAccess
- Hydra(critical) ‘TLS~BASIC’

左と同じ発行者から発行されたサーバ証明書(ターミナルサーバではない単なるWebサーバ)

```
460 590:   [3] {
464 586:     SEQUENCE {
468  14:       SEQUENCE {
470   3:         OBJECT IDENTIFIER keyUsage (2 5 29 15)
475   1:         BOOLEAN TRUE
478   4:         OCTET STRING, encapsulates {
480   2:           BIT STRING 6 unused bits
       :             '11'B
       :           }
       :         }
484  29:       SEQUENCE {
486   3:         OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
491  22:         OCTET STRING, encapsulates {
493  20:           OCTET STRING
       :             C5 01 E3 20 B1 88 03 51 7E 65 13 A8 B1 62 7D D0
       :             CC 6B D9 17
       :           }
       :         }
515  31:       SEQUENCE {
517   3:         OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
522  24:         OCTET STRING, encapsulates {
524  22:           SEQUENCE {
526  20:             [0]
       :               75 E8 03 58 5D FB 65 E4 D9 A6 AC 17 B6 03 7E 47
       :               AD 2E 81 AF
       :             }
       :           }
       :         }
548 194:       SEQUENCE {
551   3:         OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
556 186:         OCTET STRING, encapsulates {
559 183:           SEQUENCE {
562 180:             SEQUENCE {
565 177:               [0] {
568 174:                 [0] {
571  86:                   [6]
       :                     'http://tkxpasrv36.partners.extranet.microsoft.co'
       :                     'm/CertEnroll/Microsoft%20LSRA%20PA.crl'
659  84:                   [6]
       :                     'file://¥¥tkxpasrv36.partners.extranet.microsoft.'
       :                     'com¥CertEnroll¥Microsoft LSRA PA.crl'
       :                   }
       :                 }
       :               }
       :             }
       :           }
       :         }
745 305:       SEQUENCE {
749   8:         OBJECT IDENTIFIER authorityInfoAccess (1 3 6 1 5 5 7 1 1)
759 291:         OCTET STRING, encapsulates {
763 287:           SEQUENCE {
767 142:             SEQUENCE {
770   8:               OBJECT IDENTIFIER caIssuers (1 3 6 1 5 5 7 48 2)
780 129:               [6]
       :                 'http://tkxpasrv36.partners.extranet.microsoft.co'
       :                 'm/CertEnroll/tkxpasrv36.partners.extranet.micros'
       :                 'oft.com_Microsoft%20LSRA%20PA.crt'
       :               }
912 139:             SEQUENCE {
915   8:               OBJECT IDENTIFIER caIssuers (1 3 6 1 5 5 7 48 2)
925 127:               [6]
       :                 'file://¥¥tkxpasrv36.partners.extranet.microsoft.'
       :                 'com¥CertEnroll¥tkxpasrv36.partners.extranet.micr'
       :                 'osoft.com_Microsoft LSRA PA.crt'
       :               }
       :             }
       :           }
       :         }
       :       }
       :     }
       :   }
```

- keyUsage(critical) DigitalSignature Non-Repudiation
- subjectKeyID

＊＊＊ WANTED ＊＊＊

LSRA PAから発行された正式なターミナルサーバ用証明書

Flameから学ぶ証明書の安全性

- issuer/subjectUniqueId
  - 証明書偽造攻撃の格好の的
  - 使ってたら偽造証明書の可能性を疑われる
  - CSRで指定された値をそのまま流用とかはナンセンス
  - 使うのであれば、値の一意性に責任を負わなければならない
    - 耐衝突性の低いハッシュ関数を安易に使ってはダメ
- serialNumber
  - 予測困難性を甘くみてはいけない←Flameに学ぶ
  - 予測困難性を確保するには暗号技術の活用が不可欠に。。。
    - RNG, CryptoHashなど

提言

「Issuer/subjectUniqueIdentifierは使いません」宣言しましょう
「serialNumberはRNGやハッシュで予測困難性を担保しましょう