RFC 7970 : The Incident Object Description Exchange Format Version 2

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 décembre 2016

Date de publication du RFC : Novembre 2016

https://www.bortzmeyer.org/7970.html

Pour rendre plus facilement analysables les innombrables rapports d'incidents de sécurité qui circulent sur Internet tous les jours, ce RFC spécifie un format standard XML, nommé IODEF, pour décrire ces incidents. Ici, il s'agit de la version 2 de ce format IODEF, la version 1 était dans le RFC 5070 [1].

Tous les jours, des organisations comme les CERT et CSIRT, mais aussi les OIV, envoient et reçoivent des rapports détaillés concernant une attaque sur un réseau informatique ou un serveur. Ces rapports sont longs et détaillés mais, la plupart du temps, ce n'est pas une attaque isolée qui est intéressante, c'est l'image qui apparait lorsqu'on synthétise tous les rapports, et qu'on voit alors les tendances, par exemple l'arrivée d'un nouveau ver ou bien une attaque concertée contre un pays donné. D'où l'importance de pouvoir analyser automatiquement ces rapports, ce qui impose un modèle de données et un format standard, ce que fournit ce RFC.

Le modèle de données est proche des modèles objet, par exemple dans la descriptions des **classes** d'objets manipulés (comme la classe Incident en section 3.2, avec la cardinalité des attributs). Ces classes sont composés avec des données élémentaires (booléens, entiers, dates) décrites dans la section 2. Par exemple, parmi les attributs de la classe `Incident`, on trouve l'heure de début et de fin de l'incident, l'heure de détection, etc. Le schéma XML complet, écrit en W3C Schema, figure dans la section 8.

On trouve énormément de choses dans ce schéma (le RFC fait plus de 160 pages), pour traiter tous les cas prévus. Par exemple, on peut exprimer une liste de ports comprenant à la fois des ports individuels et des intervalles : `22,53,80,1024-2047`. De nombreuses classes existent pour utiliser ces informations élémentaires. Ainsi, la classe `Discovery`, une nouveauté de la version 2, permet d'indiquer comment

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5070.txt`

l'incident a été découvert (avec un attribut `source` qui a vingt valeurs possibles, comme `av` - antivirus, `os-log` - journal, `passive-dns` - un système comme DNSdb <https://www.bortzmeyer.org/dnsdb.html>, etc). Et `BusinessImpact` permet de décrire les conséquences de l'incident sur l'activité (`breach-privacy`, `loss-of-service`, `theft-financial`, etc). Ça peut même se quantifier financièrement avec la classe `MonetaryImpact`. Si on met les incidents de sécurité dans une base de données (ça s'appelle un SIEM, comme Prelude <https://www.prelude-siem.org/>), on peut donc imaginer de regarder d'abord les incidents qui ont coûté le plus cher...

Voici un exemple d'un rapport d'incident, tiré du RFC (section 7), et qui décrit et qui décrit les systèmes de C&C (quatre serveurs) d'une campagne donnée (dans le RFC 5070, l'exemple était une simple reconnaissance avec nmap...). Cet exemple a l'avantage d'illustrer la classe `IndicatorData`, une nouveauté de la version 2 :

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- A list of C2 domains associated with a campaign -->
<IODEF-Document version="2.00" xml:lang="en"
   xmlns="urn:ietf:params:xml:ns:iodef-2.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation=
   "https://www.iana.org/assignments/xml-registry/schema/
    iodef-2.0.xsd">
  <Incident purpose="watch" restriction="green">
    <IncidentID name="csirt.example.com">897923</IncidentID>
      <RelatedActivity>
        <ThreatActor>
          <ThreatActorID>
          TA-12-AGGRESSIVE-BUTTERFLY
          </ThreatActorID>
          <Description>Aggressive Butterfly</Description>
        </ThreatActor>
        <Campaign>
          <CampaignID>C-2015-59405</CampaignID>
          <Description>Orange Giraffe</Description>
        </Campaign>
      </RelatedActivity>
      <GenerationTime>2015-10-02T11:18:00-05:00</GenerationTime>
      <Description>Summarizes the Indicators of Compromise
        for the Orange Giraffe campaign of the Aggressive
        Butterfly crime gang.
      </Description>
      <Assessment>
        <BusinessImpact type="breach-proprietary"/>
      </Assessment>
      <Contact type="organization" role="creator">
        <ContactName>CSIRT for example.com</ContactName>
        <Email>
          <EmailTo>contact@csirt.example.com</EmailTo>
        </Email>
      </Contact>
      <IndicatorData>
        <Indicator>
          <IndicatorID name="csirt.example.com" version="1">
          G90823490
          </IndicatorID>
          <Description>C2 domains</Description>
          <StartTime>2014-12-02T11:18:00-05:00</StartTime>
          <Observable>
            <BulkObservable type="fqdn">
            <BulkObservableList>
              kj290023j09r34.example.com
              09ijk23jfj0k8.example.net
              klknjwfjiowjefr923.example.org
              oimireik79msd.example.org
```

```
            </BulkObservableList>
          </BulkObservable>
        </Observable>
      </Indicator>
    </IndicatorData>
  </Incident>
  </IODEF-Document>
```

Le RFC note sagement que le partage d'informations n'est pas uniquement une question technique, mais qu'elle dépend aussi des procédures bureaucratiques de chaque organisation, des contraintes légales, de la confiance (ou de l'absence de confiance, souvent justifiée) et enfin de la simple bonne ou mauvaise volonté. (Mon opinion personnelle est que, en France, le partage d'informations précises sur les incidents de sécurité est très insuffisant.)

Les changements depuis la version 1 (celle du RFC 5070) sont listés dans la section 1.4. Beaucoup de détails, beaucoup d'ajouts, parmi lesquels je note :

— Meilleure internationalisation (voir à ce sujet la section 6 du RFC), comme le fait que la classe `Contact` permette désormais d'indiquer une adresse postale en un jeu de caractères quelconque,
— Nouvelles classes (comme `IndicatorData` ou `Discovery` cités plus haut, ou comme `DomainData`, pour des informations sur un nom de domaine), et nouveaux attributs dans les classes existantes (par exemple, `Incident` y gagne `observable-id`, un identificateur qui peut être utilisé dans des références croisées).

Si l'ajout de nouvelles classes ne rendent pas les anciennes descriptions IODEF incorrectes, en revanche, certains changements cassent la compatibilité et un fichier IODEF version 1 parfait ne sera pas forcément légal pour la version 2 (cf. section 4.4). Par exemple, la sous-classe `NodeRole` (qui permet de décrire si on est attaqué par une caméra de vidéosurveillance <http://www.nextinpact.com/news/101871-dyn-on-fait-poi
htm> ou bien par un routeur <http://arstechnica.com/security/2016/11/notorious-iot-botnets-weapon
>) a changé de classe parente.

Et les mises en œuvre d'IODEF ? Un résumé de l'état de ces mises en œuvre figure dans l'"*Internet-Draft*" `draft-ietf-mile-implementreport`, et qui référence une liste des programmes IODEF <http://siis.realmv6.org/implementations/> (j'ai aussi trouvé celle-ci <http://www.ecsirt.net/service/products.html>). Parmi d'autres, on peut noter la bibliothèque de Prelude <https://github.com/Prelude-SIEM/libiodef> (et qui a une version pour l'IODEF v2 de notre RFC <https://github.com/IDMEF-IODEF/libiodefv2>), un module <http://search.cpan.org/~saxjazman/XML-IODEF-0.11/lib/XML/IODEF.pm> Perl, un autre <https://github.com/marknl/iodef> en PHP, et un troisième <http://www.decalage.info/python/iodeflib> en Python. On trouve aussi des moyens de connecter IODEF à des logiciels existants par exemple au logiciel de suivi de tâche Mantis, avec ce connecteur <https://github.com/siemens/django-mantis-iodef-importer>.

Pour des articles ou présentations sur IODEF, vous pouvez voir la "*Rump*" (session rapide) de Thomas Andrejak au SSTIC 2016 (vidéo en ligne <http://static.sstic.org/rumps2016/SSTIC_2016-06-02_P12_RUMPS_14.mp4>).

Notez en France l'existence du projet SECEF <http://www.secef.net/> ("*SECurity Exchange Format*") qui a pour objectif de promouvoir et de faciliter l[Caractère Unicode non montré [2]]usage des deux formats de fichier IDMEF (RFC 4765) et IODEF. Vous pouvez consulter leur Wiki <http://redmine.secef.net/projects/secef/wiki>, et leur tutoriel IODEF <http://redmine.secef.net/projects/secef/wiki/How_to_use_IODEF>. Il y a aussi un article de synthèse sur SECEF <http://www.silicon.fr/oiv-france-pousse-normalisation-incidents-securite-160969.html>, et un compte-rendu d'une de leurs réunions <http://www.globalsecuritymag.fr/SECEF-Day-2016-1-I
20160921,65464.html> (mais vite fait et avec des erreurs). Enfin, le RFC 8274 donne quelques conseils sur la mise en œuvre d'IODEF.

2. Car trop difficile à faire afficher par LaTeX