



HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

RÉSIST - 30 août 2004

Génération d'expressions rationnelles à partir d'événements journalisés

Denis Ducamp
<Denis.Ducamp@hsc.fr>



- x La journalisation
- x La détection d'intrusion
- x Les expressions rationnelles
- x slct
- x Des utilisations de ces expressions
- x D'autres possibilités

Plan (1/6) : la journalisation

- * La journalisation
 - * Au niveau système vs applicatif
 - * La centralisation
 - * L'analyse de journaux
 - * En temps réel vs à posteriori
 - * Mode statique vs dynamique
- * La détection d'intrusion
- * Les expressions rationnelles
- * slct
- * Des utilisations de ces expressions
- * D'autres possibilités

La journalisation

- * *Définition* : enregistrement dans des journaux d'événements générés
- * Au niveau système
 - * Une application, ou un service, génère un événement
 - * Une catégorie et une importance sont associées à cet événement
 - * L'événement est envoyé à un service de journalisation système (ex : syslogd)
 - * Ce service trie et sauvegarde l'événement
- * Au niveau applicatif
 - * L'application enregistre directement les événements qu'elle génère dans des fichiers sous son contrôle
- * La centralisation
 - * Le service de journalisation système peut renvoyer un événement à un service équivalent sur un autre système via le réseau
 - * Permet la sauvegarde et l'analyse centralisées

La journalisation : l'analyse de journaux

- * L'analyse de journaux est effectuée par des outils tiers
 - * Via l'analyse des événements enregistrés par le service de journalisation
- * Cette analyse peut être effectuée
 - * *En temps réel* : dès qu'un événement est enregistré dans un journal système
 - * *À posteriori* : les journaux sont analysés à intervalles de temps réguliers
- * Cette analyse peut être de deux types
 - * *Statique (stateless)*
 - * Chaque événement est analysé individuellement
 - * *Dynamique (stateful)*
 - * Des états, éventuellement temporaires, peuvent être créés par des événements
 - * L'analyse de certains événements peut varier suivant les états ouverts
 - * La fermeture d'un état peut générer une alerte
 - * Nécessaire à la corrélation de journaux et permet de minimiser les faux positifs.
- * L'analyse peut servir à surveiller le bon fonctionnement d'un système
 - * ainsi que réaliser de la détection d'intrusion

- 5 -

© Hervé Schauer Consultants 2003 - Reproduction Interdite



Plan (2/6) : la détection d'intrusion

- * La journalisation
- * La détection d'intrusion
 - * Au niveau système
 - * Au niveau réseau
 - * Deux méthodes théoriques
 - * Une méthode pratique
- * Les expressions rationnelles
- * slct
- * Des utilisations de ces expressions
- * D'autres possibilités

- 6 -

© Hervé Schauer Consultants 2003 - Reproduction Interdite



La détection d'intrusion

- × *Définition* : génération d'alertes à partir de l'analyse de journaux

- × Au niveau système :
 - × Analyse des événements journalisés via les fonctions systèmes
 - × Événements d'origine système ou applicative

- × Au niveau réseau :
 - × Analyse d'événements journalisés par des éléments réseau
 - × Ces éléments peuvent être des relais applicatifs, des filtres IP ou des sondes

La détection d'intrusion

- × Deux méthodes théoriques :
 - × Liste exhaustive des événements autorisés
 - × Correspond à l'application d'une politique de sécurité
 - × Une politique doit avoir été définie
 - × Nécessite une connaissance pointue du système et des applications
 - × Base de signatures décrivant des événements connus interdits
 - × Correspond à la méthode anti-virale
 - × Nécessite une mise à jour régulière, pouvant être quotidienne
 - × Nécessite une connaissance pointue du système et des applications
 - × Ces deux méthodes sont utilisées dans les logiciels commerciaux
 - × La mise à jour des bases faisant partie d'un abonnement commercial
- × Une méthode pratique :
 - × À partir des événements enregistrés, générer des motifs et les classer
 - × Nécessite beaucoup de travail initialement
 - × Peut nécessiter une mise à jour lors d'une mise à jour système ou applicative
 - × Nécessite peu de connaissance du système

HSC Plan (3/6) : les expressions rationnelles

- * La journalisation
- * La détection d'intrusion
- * Les expressions rationnelles
 - * Les chaînes fixes
 - * Les motifs
 - * posix
 - * pcre
- * slct
- * Des utilisations de ces expressions
- * D'autres possibilités

HSC Les expressions rationnelles

- * En anglais : *regular expressions*
- * Les chaînes fixes
 - * Suite fixe de caractères sans interprétation
- * Les motifs
 - * En anglais : *patterns*
 - * Des méta-caractères ont des significations spéciales
 - * ex : | * + ? { } () . \ []
 - * L'interprétation d'un motif peut ainsi représenter un grand nombre de chaînes.
 - * Deux grandes familles d'expressions rationnelles existent :
 - * *posix* : *POSIX 1003.2 regular expressions*
 - * *regex(7)*
 - * *pcre* : *Perl-compatible regular expressions*
 - * *Perlre(1)* : *perl*
 - * *pcre(3)* *pcrepattern(3)* : *libpcre*

HSC Plan (4/6) : slct

- * La journalisation
- * La détection d'intrusion
- * Les expressions rationnelles
- * slct
 - * Fonctionnement
 - * Recherches de mots et phrases
 - * Raffinement par recherche de mots variables
 - * Exemples :
 - * Une phrase et ses raffinements
 - * Scission de phrases
 - * Fusion de phrases
- * Des utilisations de ces expressions
- * D'autres possibilités

HSCslct : fonctionnement

- * slct : simple logfile clustering tool
 - * Risto Vaarandi (risto.vaarandi@eyp.ee)
 - * <http://kodu.neti.ee/~risto/slct/>
 - * Écrit en C
 - * <http://kodu.neti.ee/~risto/publications/slct-ipom03-web.pdf>
- * Recherches de mots et phrases
 - * Chaque événement est découpé en mots
 - * Les mots sont classés par nombre d'occurrences
 - * Des phrases (*clusters*) sont construites
 - * dans lesquelles les mêmes mots occupent les mêmes places
 - * Les phrases les plus courantes sont affichées
- * Raffinement par recherches de mots variables
 - * Pour tout emplacement d'une phrase sans mot fixe
 - * recherche des préfixes et suffixes communs les plus longs

HSC slct : exemples (1/3)

* Exemple :

```
$ slct -b 23 -s 100 smtpd
Mon Aug 30 10:30:51 2004: Starting...
Mon Aug 30 10:30:51 2004: Creating vocabulary...
Mon Aug 30 10:30:51 2004: 325 words inserted into the vocabulary
Mon Aug 30 10:30:51 2004: Finding frequent words from the vocabulary...
Mon Aug 30 10:30:51 2004: 5 frequent words found
Mon Aug 30 10:30:51 2004: 284 words in vocabulary occurring 1 time
Mon Aug 30 10:30:51 2004: 310 words in vocabulary occurring 2 times or less
Mon Aug 30 10:30:51 2004: 318 words in vocabulary occurring 5 times or less
Mon Aug 30 10:30:51 2004: 318 words in vocabulary occurring 10 times or less
Mon Aug 30 10:30:51 2004: 318 words in vocabulary occurring 20 times or less
Mon Aug 30 10:30:51 2004: Finding cluster candidates...
Mon Aug 30 10:30:51 2004: 2 cluster candidates found
Mon Aug 30 10:30:51 2004: Finding clusters from the set of candidates...
Mon Aug 30 10:30:51 2004: 1 clusters found
* * to=<user@toyuuucp.hsc.fr>, relay=uucp, delay=1, status=sent (toy)
Support: 107

Mon Aug 30 10:30:51 2004: Analysis complete
```

- 13 -

© Hervé Schauer Consultants 2003 - Reproduction Interdite



HSC slct : exemples (2/3)

* Le raffinement par recherche de mots variables

```
$ slct -b 23 -s 100 -r smtpd
postfix/pipe[*]: *: to=<user@toyuuucp.hsc.fr>, relay=uucp, delay=1,
status=sent (toy)
Support: 107
```

* La réécriture en expression rationnelle, ici *pcr*

```
pcrereg 'postfix/pipe\[.*\]: .*: to=<user@toyuuucp.hsc.fr>, relay=uucp,
delay=1, status=sent \(\toy\) ' smtpd | wc
107 1177 12900
```

* Le raffinement manuel

▸ Remplacement du pid

```
pcrereg 'postfix/pipe\[d+\]: .*: to=<user@toyuuucp.hsc.fr>, relay=uucp,
delay=1, status=sent \(\toy\) ' smtpd | wc
107 1177 12900
```

▸ Remplacement des spécificités

```
pcrereg 'postfix/pipe\[d+\]: .*: to=<user@toyuuucp.hsc.fr>, relay=uucp,
delay=d+, status=sent \(\toy\) ' smtpd | wc
182 2002 21944
pcrereg 'postfix/pipe\[d+\]: [[:xdigit:]]*: to=<\S+>, relay=uucp,
delay=d+, status=sent \(\S+\) ' smtpd | wc
182 2002 21944
```

- 14 -

© Hervé Schauer Consultants 2003 - Reproduction Interdite



HSC slct : exemples (3/3)

- x Scissions de phrases
 - x Il est possible qu'il soit avantageux, dans le cas de corrélations d'événements, de séparer en deux phrases distinctes une phrase proposée par slct
 - x Ex :
 - x `sshd[\d+]: (Accepted|Failed) (password|publickey) for \S+ from \S+ port \d+ (ssh2)?`
- x Fusion de phrases
 - x Il est possible qu'il soit avantageux, dans le cas d'amélioration des performances, de fusionner en une seule phrases deux phrases proposées par slct
 - x Ex :
 - x `postfix/smtp[\d+]: TLS connection established to \S+: (TLSv1|SSLv[23]) with cipher \S+ \(\d+/\d+ bits\)`
 - x `postfix/smtpd[\d+]: TLS connection established from \S+[\.\d+]: (TLSv1|SSLv[23]) with cipher \S+ \(\d+/\d+ bits\)`

HSC Plan (5/6) : des utilisations de ces expressions

- x La journalisation
- x La détection d'intrusion
- x Les expressions rationnelles
- x slct
- x Des utilisations de ces expressions
 - x Scripts maison
 - x La classification des expressions : *alert, ignore, unknown*
 - x *egrep(1) / pcregrep(1)*
 - x Affichage temps réel : *colortail...*
 - x Mode statique : *swatch, logcheck...*
 - x Mode dynamique : *logsurfer, sec...*
- x D'autres possibilités

HSC Des utilisations de ces expressions : scripts maison

- * La classification des expressions
 - * Les expressions rationnelles doivent être classées en deux groupes :
 - * *alert*, *ignore*
 - * Tout événement inconnu est suspect et donc reporté (*unknown*)
 - * Cette classification permet une utilisation correcte
 - * En suivant cet ordre : *alert*, *ignore*, *unknown*
 - * L'ordre *ignore*, *alert*, *unknown* peut être trop permissif, et sujet à erreurs
 - * mais légitime dans certains cas
- * *egrep(1)* / *pcgrep(1)*
 - * Les journaux sont filtrés via les commandes *egrep* ou *pcgrep*
 - * En utilisant leur option *-f*
 - * Il peut être nécessaire de modifier le code de ces commandes si elles possèdent une limite en nombre d'expressions rationnelles par fichier (100 pour *pcgrep*)
 - * Un ou plusieurs fichiers temporaires peuvent être nécessaires pour trier les événements suivant le nombre de niveaux d'alerte

- 17 -

© Hervé Schauer Consultants 2003 - Reproduction Interdite



HSC Des utilisations... pour l'affichage temps réel

- * *colortail*
 - * Joakim Andersson
 - * <http://www.student.hk-r.se/~pt98jan/colortail.html>
 - * Écrit en C
 - * Fonctionne comme *tail* avec colorisation de motifs
 - * Plusieurs motifs peuvent être reconnus par ligne
 - * Utilise les expressions rationnelles *posix*.
 - * Exemple :

```
    COLOR brightred
    {
    # matches the word "root"
    ^.*(root).*$
    }
    $ colortail -f -k config.file log.file
```

- 18 -

© Hervé Schauer Consultants 2003 - Reproduction Interdite



HSC Des utilisations... en mode statique : swatch

- * *swatch*
 - * Todd Atkins
 - * <http://swatch.sourceforge.net/>
 - * Écrit en perl
 - * Associe à chaque motif une ou plusieurs actions
 - * *echo [couleur], bell [n], mail [adresse], write [user[:user...]]*
 - * *exec commande, pipe commande, continue, quit*
 - * Exemple :

```
watchfor  /INVALID|REPEATED|INCOMPLETE/  
          echo inverse  
          bell 3  
ignore   /sendmail/,/nntp/,/xntp|ntpd/,/faxspooler/  
$ swatch --config-file=~/.swatchrc --tail-file=/var/log/messages
```

HSC Des utilisations... en mode statique : logcheck

- * *logcheck*
 - * Debian Logcheck Team
 - * <http://packages.debian.org/unstable/admin/logcheck>
 - * Écrit en perl
 - * Associe les événements journalisés à une série d'ensembles de motifs *posix*
 - * Envoie le résultat à l'administrateur
 - * Équivalent à la solution maison
 - * Niveaux : ignore (paranoid, server, workstation), *cracking, violation*
 - * S'exécute à chaque redémarrage du système et à chaque heure
 - * Sans retraiter deux fois le même événement
 - * Sous debian certains packages fournissent leurs propres ensembles de motifs

HSC Des utilisations... en mode dynamique : logsurfer

- * *logsurfer*
 - * Wolfgang Ley and Uwe Ellerman
 - * <http://www.cert.dfn.de/eng/logsurf/>
 - * Écrit en C
 - * Le premier analyseur gérant des contextes
 - * Peut analyser tout fichier texte, pas seulement des journaux syslog
 - * Exemple :

```
' ([^ ]*) xntpd\[([0-9]*)\]: synchronisation lost' - - - 0 CONTINUE
  open " $2 xntpd\[[$3\]:" - 100 3600 0
  pipe "/usr/lib/sendmail root"
' ([^ ]*) xntpd\[([0-9]*)\]: synchronisation lost' - - - 0
  rule before
  " ($2) xntpd\[($3\): synchronized to" - " ($2) xntpd\[
[$3\]: synchronized to" - 3600
  delete " $2 xntpd\[[$3\]:"
```

HSC Des utilisations... en mode dynamique : sec

- * *sec* : simple event correlation
 - * Risto Vaarandi (risto.vaarandi@eyp.ee)
 - * <http://www.estpak.ee/~risto/sec/>
 - * Écrit en perl
 - * Documentation et exemples :
 - * <http://kodu.neti.ee/~risto/publications/sec-ipom02-web.pdf>
 - * <http://sixshooter.v6.thrupoint.net/SEC-examples/article.html>
 - * <http://sixshooter.v6.thrupoint.net/SEC-examples/article-part2.html>
 - * Exemple :

```
type=SingleWithThreshold
ptype=RegExp
pattern=user (\S+) login failure on (\S+)
desc=Repeated login failures for user $1 on $2
action=shellcmd notify.sh "%s"
window=60
thresh=3
```

Plan (6/6) : d'autres possibilités

- x La journalisation
- x La détection d'intrusion
- x Les expressions rationnelles
- x slct
- x Des utilisations de ces expressions
- x D'autres possibilités
 - x Les journaux applicatifs

D'autres possibilités

- x Les journaux applicatifs
 - x Les journaux des serveurs web sont les plus courants
 - x Nécessitent un pré-traitement pour
 - x extraire la requête
 - x la décoder (%xx) et
 - x séparer les variables et les valeurs des noms de scripts
 - x Certainement beaucoup d'autres...
 - x À voir au cas par cas
 - x Le préfixe à ignorer doit être de longueur constante
 - x Définir l'ensemble des caractères correspondant aux séparateurs de champs
 - x Espaces par défaut

Conclusion

- × Le travail le plus fastidieux est facilité
 - × Les événements les plus journalisés sont rapidement repérés
 - × Les événements les plus rares peuvent être traités au jour le jour

- × Il faut savoir dans quel(s) programme(s) les expressions rationnelles seront utilisées
 - × Pour choisir le type de motifs (*pcre* ou *posix*)
 - × Pour classer les expressions

- × Il faut ensuite consulter les résultats
 - × Ce qui permet d'effectuer de la détection d'intrusion
 - × Mais surtout de surveiller le bon fonctionnement du système

Questions...

Merci de votre attention

Et n'hésitez pas à me poser vos questions...

...sans réveiller ceux qui se sont endormis ;-)