

# Une approche pragmatique de la mise en œuvre de politiques de sécurité

**Guillaume Rousse**

INRIA, DSI

**Denis Joiret**

INRIA, DSI

**Bertrand Wallrich**

INRIA

## Résumé

*Il y a deux catégories d'administrateurs système et réseau :*

- *ceux qui écrivent des politiques de sécurité ;*
- *ceux qui sont censés les mettre en œuvre.*

*Comme les premiers ont tendance à écrire plus vite que les seconds ne mettent en œuvre, il vaut mieux pouvoir mesurer l'écart entre la théorie et la pratique, et ce afin de chercher le meilleur moyen de faire converger les deux.*

*Cet article présente l'approche que nous utilisons à l'INRIA pour évaluer la conformité de nos ressources informatiques vis-à-vis de notre politique de sécurité. A partir de données brutes fournies par différentes sources, nous extrayons les éléments qui nous intéressent, et nous mettons en évidence les écarts avec nos préconisations. Le résultat est alors présenté sous la forme d'un rapport synthétique, faisant le lien entre la règle du jeu (la politique) et la réalité. Au delà de l'aspect purement technique de cette recherche de vulnérabilités, ce sont des considérations humaines qui président aux choix des indicateurs, au langage, à la présentation des résultats, et à la volonté de présenter ceux-ci dans notre contexte spécifique, avec des solutions techniques prêtes à l'emploi. Identifier les problèmes, c'est bien, les faire corriger, c'est encore mieux.*

## Mots clefs

*politique de sécurité, conformité*

## 1 Introduction

Il y a deux catégories d'administrateurs système et réseau :

- ceux qui écrivent des politiques de sécurité ;
- ceux qui sont censés les mettre en œuvre.

Dans la vraie vie, la deuxième catégorie est généralement débordée par de multiples tâches plus urgentes, et manque de temps à consacrer au sujet. Du coup, les recommandations restent sans effet, s'accumulent, et l'écart entre la théorie et la pratique tend à croître indéfiniment...

Il est donc nécessaire de pouvoir évaluer cet écart, afin d'en prendre conscience d'une part, et de chercher à le réduire d'autre part. En langage de décideur pressé, mettre en place des indicateurs, et s'en servir pour piloter les évolutions.

## 2 Des outils...

### 2.1 Principe

Dans le domaine de la sécurité, il existe une multitude d'outils d'analyse. Ceux-ci sont capables de fournir une grande quantité de données brutes, mais celles-ci ne correspondent pas forcément à l'information recherchée.

Par exemple, un scanner de port permet de rechercher les services accessibles sur une machine. Mais il est incapable de distinguer ceux qui sont légitimes de ceux qui ne le sont pas, parce qu'il ne s'agit plus d'une information technique, mais contextuelle. Un scanner de vulnérabilité va identifier un certain nombre de failles, potentielles ou avérées, sur une machine, et leur affecter une gravité. Mais celle-ci est relativement arbitraire : l'utilisation d'une distribution Linux dont la période de support officielle est terminée (risque critique pour Nessus) constitue-t-elle vraiment un risque plus élevé qu'un accès SSH lorsque l'authentification par mot de passe est autorisée (simple information pour Nessus) ? Là encore, il ne s'agit pas d'un critère technique, mais d'une évaluation subjective du risque posé par deux menaces différentes.

Bref, ces résultats bruts ne permettent pas de vérifier directement une politique de sécurité, c'est-à-dire un ensemble de préconisations établies en fonction de ce que l'on cherche à protéger, contre qui, et comment. Par contre, il est possible de traiter ces résultats, de façon à faire ressortir les infractions à cette politique. Mieux encore, il est possible de mesurer ces écarts, de suivre l'évolution de ceux-ci avec le temps, et donc de produire une mesure de conformité de la réalité par rapport à la politique de sécurité.

### 2.2 Réalisation

Nous avons construit un outil permettant d'arriver à cet objectif, en utilisant un scanner de vulnérabilité réputé, Nessus[1], comme source de données brutes et un peu de programmation autour. Cet outil a reçu le petit nom de Solution de Test Automatisée de la Sécurité Informatique, ou STASI pour les intimes. Son fonctionnement est illustré par la figure 1.

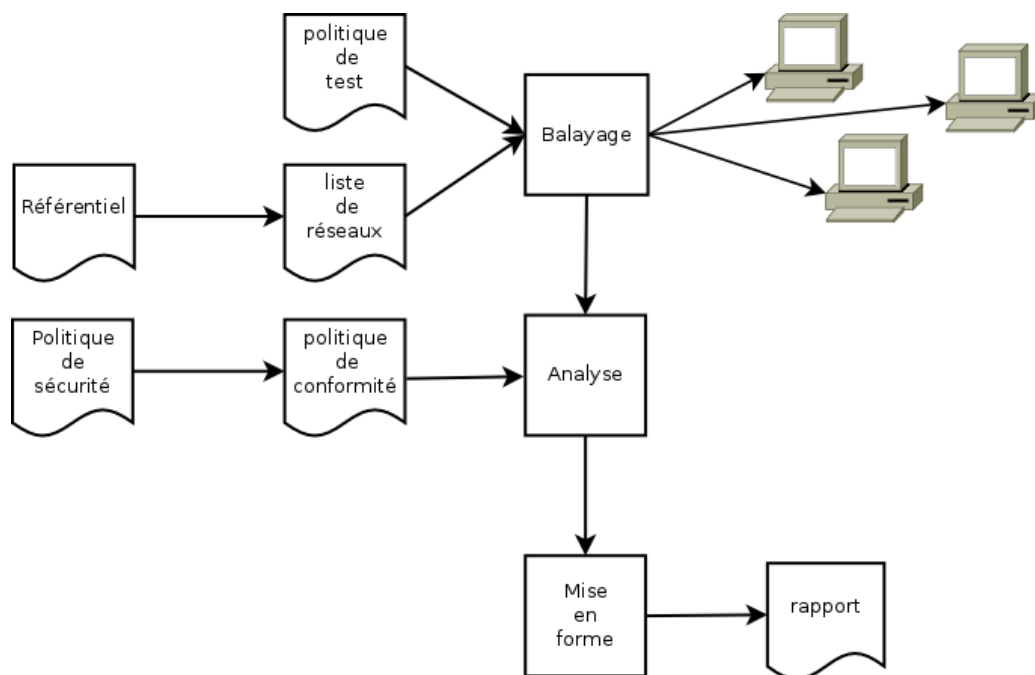


Figure 1 - Schéma de fonctionnement

L'outil prend en entrée une liste de réseaux et une politique de test. La liste de réseaux est un fichier de configuration pour l'outil, produit à partir d'un référentiel, qui énumère et caractérise les différents réseaux à tester. La politique de test est un objet interne à Nessus, qui définit les tests à réaliser. L'outil pilote alors Nessus via son interface RPC pour balayer ces différents réseaux un par un.

L'outil récupère et analyse ensuite les résultats produits, en fonction d'une politique de conformité, pour identifier les violations. Cette politique de conformité est un fichier de configuration, qui traduit sous une forme opérationnelle pour

notre outil notre politique de sécurité qui, elle, s'adresse aux humains. Par exemple, une préconisation qui s'énonce «pas de service accessible depuis l'extérieur sur les réseaux de postes de travail» se vérifie en cherchant la présence d'un port ouvert sur ce réseau. Une préconisation qui s'énonce «pas d'authentification HTTP sur une connexion non sécurisée» se vérifie en recherchant la présence de résultat pour le plugin Nessus n°34850. Et le nombre de violations de chacune de ces préconisations constitue alors un indicateur numérique de conformité d'une ressource (une plage d'adresses IP ici) à notre politique de sécurité.

Enfin, l'outil construit à partir de ces indicateurs un rapport synthétique, sous la forme d'un ensemble de pages web, qui montre l'évolution des valeurs au cours des semaines écoulées. Les résultats bruts des outils sont également mis à disposition, pour plus d'information. Les captures d'écran des figures 2 et 3 montrent l'exemple d'un tel rapport.

| plage                             | nom            | classe | sous-classe       | accessibilité | vulnérabilité | conformité |
|-----------------------------------|----------------|--------|-------------------|---------------|---------------|------------|
| <a href="#">129.94.129.0/24</a>   | production     | fermé  | services internes | OK            | OK            | OK         |
| <a href="#">129.94.131.0/24</a>   | administration | fermé  | administration    | OK            | OK            | OK         |
| <a href="#">129.94.143.0/24</a>   | numérique      | fermé  |                   | OK            | OK            | OK         |
| <a href="#">129.94.145.0/24</a>   | tma            | fermé  |                   | OK            | OK            | OK         |
| <a href="#">129.94.163.0/23</a>   | dmz            | ouvert | services externes | OK            | OK            | KO         |
| <a href="#">129.94.185.0/24</a>   | vpn-ng roc     | fermé  |                   | OK            | OK            | OK         |
| <a href="#">193.94.123.64/27</a>  | bbnat2         | fermé  | interco           | OK            | OK            | OK         |
| <a href="#">193.94.123.96/27</a>  | bbnat1         | fermé  | interco           | OK            | OK            | OK         |
| <a href="#">194.52.194.0/25</a>   | qualification  | fermé  |                   | OK            | OK            | OK         |
| <a href="#">194.52.194.128/25</a> | dmz            | ouvert | services externes | OK            | OK            | KO         |
| <a href="#">194.52.197.0/25</a>   | ressources     | fermé  |                   | OK            | OK            | OK         |

Page générée le 12/11/2013 à 11:14:44

Figure 2 - Rapport synthétique : vue d'ensemble

### 129.94.163.0/23

**Information**

description: dmz  
classe: ouvert  
sous-classe: services externes

**Résultats synthétiques**

- Horaires
- Accessibilité: OK
- Vulnérabilité: OK
- Conformité: KO: 13 erreur(s)

| indicateur                                       | valeur | résultat actuel<br>détails | historique |       |       |
|--|--------|----------------------------|------------|-------|-------|
|  |        |                            | 03/11      | 27/10 | 20/10 |
| certificats SSL émanant d'une autorité inconnue  | 0      |                            | 0          | 0     | 0     |
| certificats SSL auto-signés                      | 0      |                            | 0          | 0     | 0     |
| certificats SSL expirés                          | 0      |                            | 0          | 0     | 0     |
| utilisations du protocole SSLv2                  | 0      |                            | 0          | 0     | 0     |
| utilisations de suites cryptographiques anonymes | 3      | • 129.94.163.28:443        | 3          | 3     | 3     |
| utilisations de suites cryptographiques faibles  | 2      | • 129.94.163.28:443        | 2          | 2     | 2     |
| utilisations de suites cryptographiques moyennes | 3      | • 129.94.163.4:443         | 3          | 3     | 3     |
| disponibilité des méthodes TRACE/TRACK           | 2      | • 129.94.163.62:80         | 2          | 2     | 4     |
| vulnérabilités RANGE                             | 0      |                            | 0          | 0     | 0     |
| authentifications HTTP en clair                  | 1      | • 129.94.163.62:80         | 1          | 1     | 2     |
| authentifications par formulaire en clair        | 2      | • 129.94.163.25:80         | 3          | 3     | 3     |
| bannière trop détaillées                         | 0      |                            | 0          | 0     | 0     |
| authentification par mot de passe                | 0      |                            | 0          | 0     | 0     |
| utilisation du protocole SSHv1.0                 | 0      |                            | 0          | 0     | 0     |

**Résultats bruts**

[Machine](#)  
[Vulnérabilité](#)

Page générée le 12/11/2013 à 11:14:44

Figure 3 - Rapport synthétique : vue d'un réseau

L'outil n'étant pas interactif, son utilisation peut être automatisée. Ainsi, l'ensemble des plages réseaux de l'INRIA est analysé automatiquement chaque semaine, à partir d'une machine hébergée chez un prestataire externe, en dehors des heures d'activité.

## 2.3 Réglages

Nessus, notre source de données, est relativement trivial à utiliser. Il suffit de définir une cible et une politique de test, et de le lancer. Toutefois, l'usage montre rapidement qu'il est nécessaire de se plonger un minimum dans les différentes options disponibles pour éviter certains problèmes.

D'abord, balayer l'ensemble des adresses d'un réseau peut être long. Notamment si les filtrages mis en place ne permettent pas de distinguer immédiatement une machine qui ne répond pas d'une adresse inoccupée. C'est sans doute un problème de nant, mais avec les quatre réseaux de 65000 adresses publiques de l'INRIA, il arrive fréquemment que le test d'un site ne soit pas terminé lorsque le suivant commence, même avec vingt-quatre heures de décalage entre les deux...

Ensuite, balayer un réseau, cela peut également être intrusif. À tort ou à raison, les collègues identifient rapidement l'outil de test comme le coupable idéal pour toute perturbation constatée...

La première mesure de remédiation consiste à annoncer le planning de test, et à faire figurer dans les résultats les heures de début et de fin de chaque passage de l'outil, de façon à rapidement mettre en évidence l'absence de responsabilité. Et le suivi de la durée de chaque passage permet également de mesurer l'impact des variations de configuration.

La seconde mesure consiste à limiter le nombre de connexions simultanées et de paquets par seconde. Mobiliser trop de ressources peut amener à un déni de service sur les routeurs d'entrée de site, notamment vers des réseaux NATés (mais met en évidence une fragilité sous-jacente), ou au déclenchement de mécanismes de protection contre des attaques de type SYN-flood. À l'inverse, utiliser trop peu de ressources revient à retomber dans le problème de durée précédent. Il y a donc un véritable travail d'équilibrage de charge, à réaliser avec les équipes qui gèrent les équipements réseaux.

Enfin, la troisième mesure consiste à limiter la plage de ports et la liste des vulnérabilités à tester. Ce qui revient à faire un compromis entre exhaustivité d'une part, limitation de l'impact et rapidité d'autre part. Dans l'optique d'une procédure automatisée hebdomadaire à large spectre, et étant donné que la liste des résultats trouvés suffit déjà largement à nous occuper, nous avons rapidement opté pour la limitation. Vu le peu de valeur ajoutée d'une sélection fine des tests à lancer par Nessus, qui gère de toute façon des dépendances internes, cette limitation consiste à désactiver en bloc certaines catégories de tests (tests de déni de service, notamment) ou certaines options de configuration globale (tests de paramètres web).

## 2.4 Évolutions

À l'origine, nous utilisions également un scanner de port, Nmap[2], mais très rapidement la duplication de fonctionnalités entre les deux (Nessus passe également par une phase de recherche de services) nous a conduit à abandonner son usage. Par contre, le principe de l'utilisation de plusieurs sources de données reste conservée dans l'outil.

La principale limitation technique constatée aujourd'hui est le manque d'adéquation de Nessus vis-à-vis de la recherche de failles dans des applications web, lors de la recherche sur des réseaux complets. En effet, l'outil considère alors une liste d'adresses IP, et ne va tester que l'hôte virtuel HTTP correspondant à l'adresse IP de chaque machine, bien souvent l'hôte virtuel par défaut. Et même si celui-ci est le seul présent, l'outil est incapable avec une politique généraliste de détecter un site qui n'est pas atteignable depuis la racine du serveur web. Pour cela, il faut utiliser une politique dédiée. Bref, l'outil est adapté à un balayage en profondeur (une seule machine cible, avec une politique de test dédiée) ou en largeur (un réseau complet, avec une politique de test généraliste), alors que la cible ici est intermédiaire. Accessoirement, il faudrait aussi être capable de fournir la liste des noms d'hôtes à tester, ce qui n'est pas une mince affaire en l'absence d'un référentiel...

## 3 ...et des hommes

Identifier les problèmes, c'est bien, les corriger, c'est encore mieux. Or nous nous situons précisément dans une logique de séparation des rôles, dans laquelle nous n'avons pas la responsabilité ni la possibilité d'intervenir sur les équipements concernés. Il ne s'agit donc plus de corriger, mais de faire corriger, ce qui change un certain nombre de choses, et notamment le type de stratégie à mettre en œuvre.

### 3.1 Présentation des résultats

Dès qu'il s'agit de communiquer avec un tiers, la forme joue autant que le fond. Nul besoin d'être un expert en communication pour définir quelques points importants.

D'abord, mieux vaut produire des rapports ciblés, limités aux seules ressources gérées par l'interlocuteur visé. Par exemple, inutile de noyer l'administrateur d'une machine unique avec des informations concernant l'ensemble des machines situées sur le même réseau. Dans notre contexte de répartition des rôles, avec une équipe gérant l'infrastructure d'une part, une autre équipe gérant les services d'autre part, c'est malheureusement souvent un problème complexe. La qualité du référentiel utilisé pour définir les cibles joue ici un rôle primordial.

Ensuite, mieux vaut adapter le rapport à son public, notamment en choisissant un vocabulaire et un niveau de détail correspondant au niveau de compétence supposé de son interlocuteur. Celui-ci est généralement variable, mais il ne s'agit clairement pas d'experts en sécurité. Donc mieux vaut éviter de supposer par exemple que les arcanes du protocole TLS sont maîtrisés...

Enfin, mieux vaut mettre en évidence ce qui doit être corrigé, pourquoi, et comment. Dans notre cas, ceci se traduit par une volonté de faire le lien entre :

- l'occurrence du problème (un service accessible de l'extérieur présente un certificat auto-signé) ;
- le lien avec nos préconisations (tous les services en production doivent utiliser des certificats fournis via l'offre TCS de Renater) ;
- les différentes façons possible de le corriger (faire une demande de certificat, ou fermer l'accès au service depuis l'extérieur).

Au final, il s'agit d'un savant équilibre entre concision et justification. Comme il s'agit d'un sujet hautement subjectif, une méthode possible consiste à se mettre à la place de son interlocuteur, d'imaginer tous les arguments utilisables pour justifier l'inaction (« je ne comprend pas ce dont il s'agit », « où est la liste des machines à corriger ? », etc.), et de s'attacher à rendre ces arguments inutilisables...

### 3.2 Lien avec la politique de sécurité

Le résultat brut d'un outil de détection de vulnérabilités, c'est... une liste de vulnérabilités. Il faut ensuite filtrer cette liste pour éliminer les faux positifs, exclure les vulnérabilités considérées comme acceptables, puis ordonner celles qui restent pour définir des priorités. N'importe quel administrateur système ayant déjà utilisé ce type d'outil est habitué à faire ainsi son propre tri dans les résultats fournis, sur ses propres critères. Et il va naturellement faire de même ici en présence d'un nouveau rapport, surtout s'il n'est pas à l'origine de la demande.

Les résultats produits ici sont différents, parce que tout ce travail de post-traitement des résultats bruts a déjà été fait. Les critères de sélection et de tri sont explicites d'une part, ce qui les rend explicables, basés sur la politique de sécurité de l'établissement d'autre part, ce qui les légitimise. Bref, il faut expliquer qu'il ne s'agit pas d'une liste de vulnérabilités potentielles, mais bien d'une liste d'erreurs de conformité avérées, et que cela est censé suffire à justifier leur correction. La politique de sécurité, et les préconisations de sécurité qui en découlent, constituent une règle du jeu, et l'outil ne fait que vérifier son application.

Néanmoins, l'argument d'autorité a ses limites, et il est relativement illusoire d'espérer obtenir beaucoup de résultats par simple rappel des règles. Surtout si ces règles sont arbitraires, et que l'obéissance au règlement ne fait pas vraiment partie de la culture de nos établissements de recherche... Il faut donc pouvoir faire évoluer la politique de sécurité, s'il s'avère que certaines de ses préconisations ne sont pas pertinentes, et inapplicables. Dans ce sens, l'outil fournit finalement une boucle de rétroaction, en confrontant une politique à la réalité. Tant que cette politique reste un document abstrait, cet ajustement est plus difficile.

L'approche pragmatique consiste alors à considérer que pour réduire la distance entre la théorie (le niveau de sécurité visé) et la pratique (le niveau de sécurité actuel), il y a deux leviers possibles :

- augmenter le niveau de sécurité actuel ;
- diminuer le niveau de sécurité visé.

Autrement dit, diminuer ses objectifs, pour les atteindre plus facilement. Quitte à les rehausser ensuite, dans un deuxième temps, pour progresser par étape. Cette approche itérative, et la recherche d'un compromis réalisable, rejoignent le concept de sécurité homéopathique formulé dans [3], et auquel toute notre approche doit beaucoup : une sécurité savamment administrée, à petites doses, afin de réduire les risques auxquels s'expose une entité donnée dans le temps, à travers un processus d'amélioration continue.

À contrario, l'approche cynique consiste à considérer que si le niveau atteint correspond à une fraction donnée du niveau visé, il faut plutôt viser plus haut que nécessaire pour espérer atteindre l'objectif initial... L'outil, lui, est complètement agnostique, et ne fait que fournir les indicateurs nécessaires dans un cas comme dans l'autre.

### 3.3 Appropriation de l'outil

Tel que nous l'utilisons, l'outil n'est pas un service à disposition de nos administrateurs de ressources informatiques. Ils n'ont guère de latitude sur les cibles de nos tests (tous les réseaux accessibles de l'extérieur), ni sur le type d'informations que nous recherchons (les violations de notre politique). Autrement dit, il n'y a pas forcément de besoin exprimé, ni d'adhésion spontanée à la démarche. Et pourtant, nous avons besoin de leur participation... Il va donc falloir s'attacher à susciter un minimum d'appropriation.

Pour commencer, il suffit d'être à l'écoute des premiers retours, et d'en tenir compte pour améliorer l'outil. Nous avons ainsi par exemple amélioré la présentation des rapports, et nous travaillons actuellement à plus de flexibilité dans la granularité et le rythme de passage.

Ensuite, il semble judicieux de faciliter l'investissement, par la sélection des indicateurs mis en avant. En considérant la simplicité de correction, et pas uniquement la gravité d'un problème, il est possible de donner à tout le monde la possibilité d'améliorer le niveau de sécurité actuel du Système d'Information, en commençant par des choses simples. Typiquement, mettre en place un certificat X.509 correct, lorsque la procédure est documentée, est à la portée de n'importe quelle personne gérant un serveur, et s'inscrit pleinement dans la logique de compromis présentée plus haut.

Enfin, le fait de conserver un historique des tests effectués permet également d'utiliser l'outil comme un test de non-régression en cas de modification de configuration. Nos collègues administrateurs réseaux, en particulier, ont apprécié la possibilité de comparer un indicateur du nombre de machines accessibles depuis l'extérieur avant et après le changement des équipements de filtrage d'un site.

## 4 Conclusion

Une politique de sécurité n'apporte guère d'intérêt si elle n'est pas appliquée. Il faut donc pouvoir mesurer la différence entre la théorie et la pratique, pour ajuster à la fois le niveau actuel et l'objectif poursuivi. Après tout, une politique moins ambitieuse mais mieux suivie est parfois plus efficace qu'une politique draconienne complètement ignorée... Encore une fois, la sécurité informatique est un savant compromis entre des possibilités techniques et des contraintes humaines et organisationnelles.

## Bibliographie

- [1] [http://fr.wikipedia.org/wiki/Nessus\\_\(logiciel\)](http://fr.wikipedia.org/wiki/Nessus_(logiciel)).
- [2] <http://fr.wikipedia.org/wiki/Nmap>.
- [3] Saad Kadhi. Réduction de la surface d'attaque d'un S.I. : une approche pragmatique. *MISC*, (58) :75–82, Novembre 2011.