Bro : un NIDS pas comme les autres ?

Blabla : G. Arcas
Démo : S. Kadhi
Contributions : J.P Luiggi – F. Debieve
Réunion du groupe SUR/OSSIR 16 janvier 2007

Présentation du projet Bro

- Bro c'est :
  - YALBNIDS (Yet Another Libpcap Based NIDS)
  - Destiné à la détection sur des réseaux Gbps à fort trafic.
  - Projet lancé en 1998 par Vern Paxson (ICIR/ICSI/LBNL)
    - http://bro-ids.org
- Conçu pour :
  - Minimiser sinon éviter toute perte de paquet
  - Minimiser sinon éviter tout faux-positif
  - Résister aux attaques dirigées contre la sonde
  - Permettre une remontée d'alerte en temps réel
  - Accepter facilement l'ajout de nouvelles fonctionnalités
- Fonctionnel mais encore expérimental.
    - Version Développement : 1.2.1 (octobre 2006)
    - Version stable : 1.1d

Vern Paxson

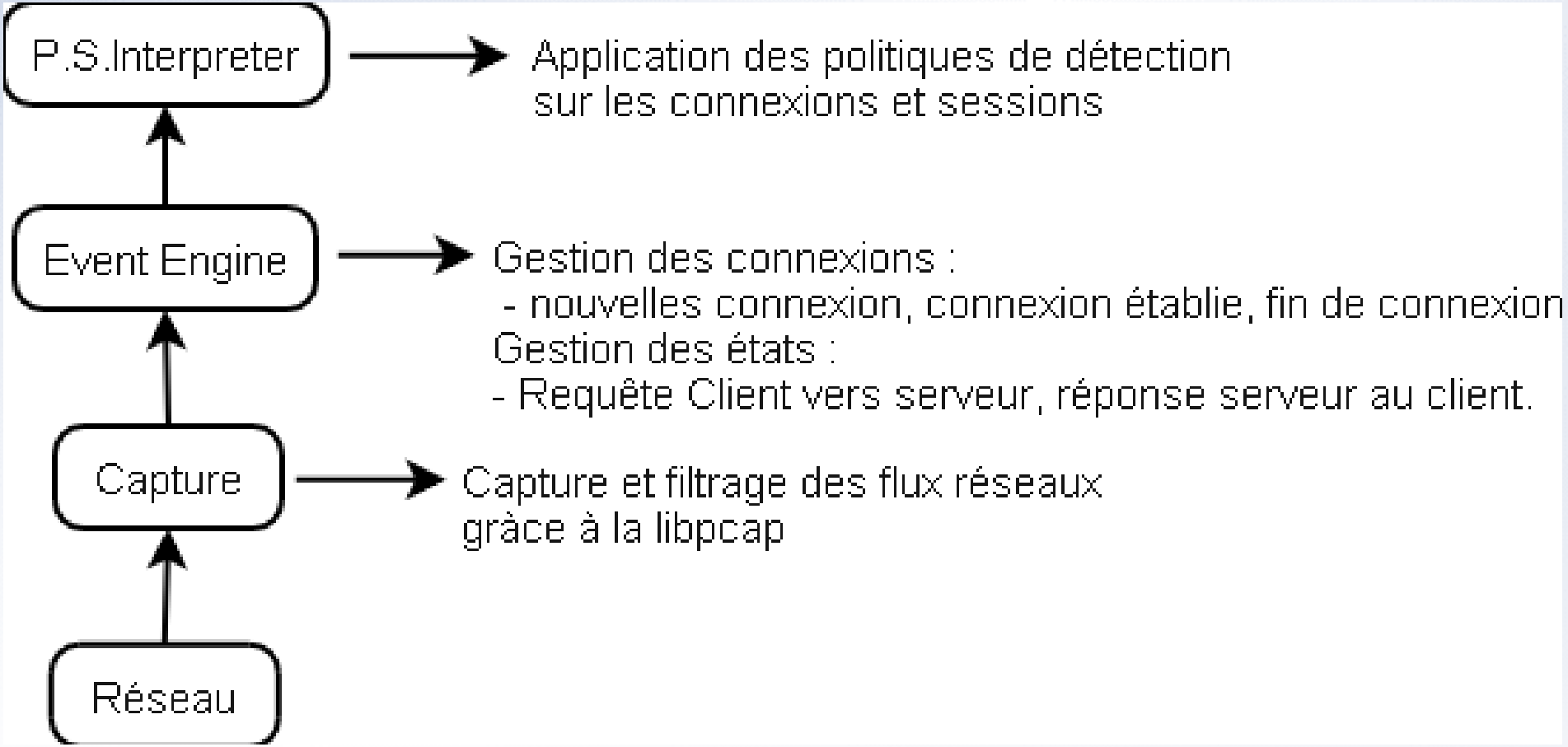
- 1998
  - Bro: A System for Detecting Network Intruders in Real-Time.
- 1999
  - Known TCP Implementation Problems, RFC 2525
  - TCP Congestion Control, RFC 2581
- 2000 - 2001
  - Detecting Backdoors, Detecting Stepping Stones
  - Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics
    - ➔ scrub (nettoyage du trafic dans PF/*BSD)
- 2002 - 2006
  - How to 0wn the Internet in Your Spare Time, A Taxonomy of Computer Worms, Inside the Slammer Worm, The Spread of the Sapphire/Slammer Worm, The Top Speed of Flash Worms, Very Fast Containment of Scanning Worms, A Worst-Case Worm
  - Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection

- Maitrise donc le sujet ! :-)

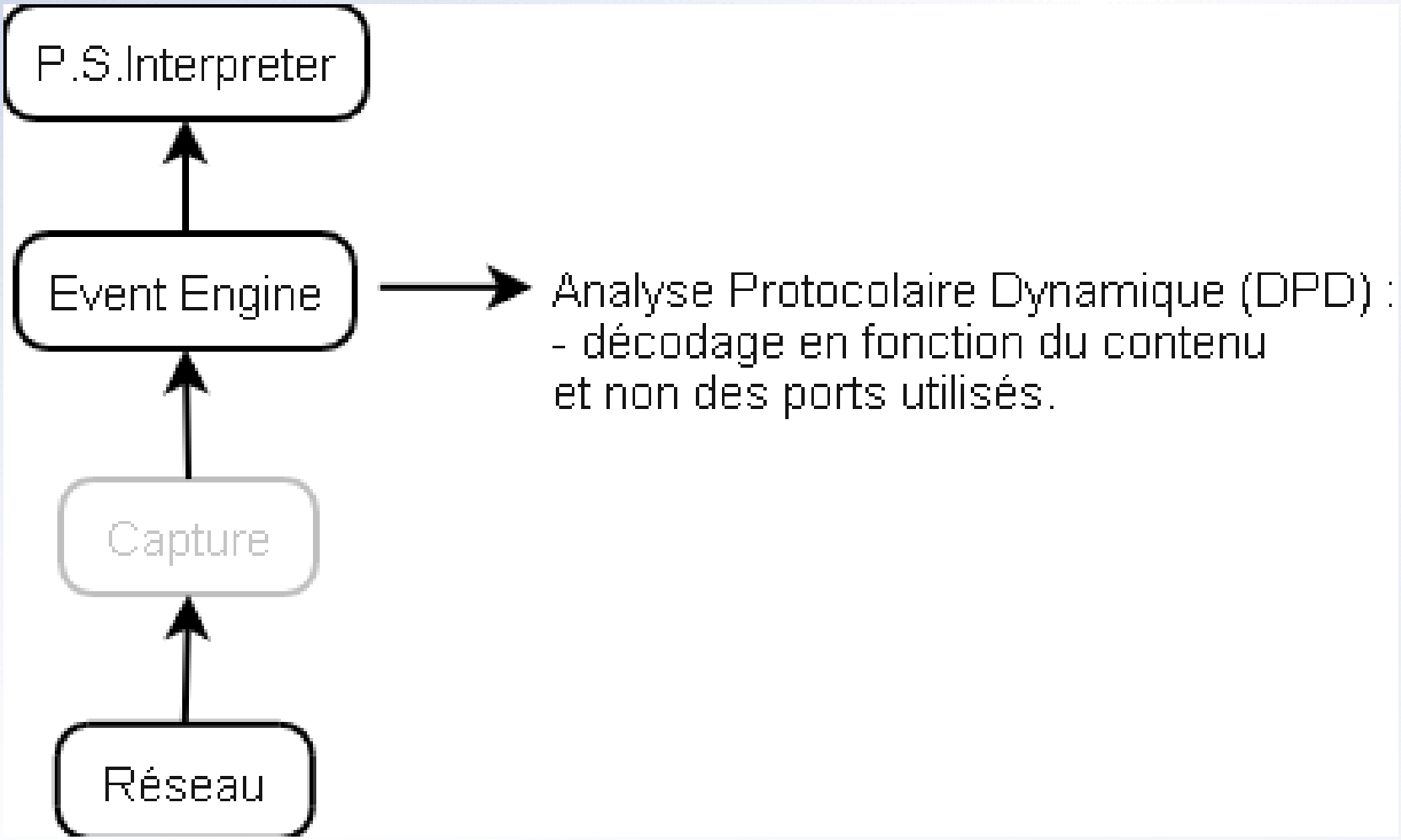
Caractéristiques

- Trois grandes familles de NIDS :
- Détection par signatures
  - Une signature décrit les caractéristiques d'un paquet ou d'un évènement jugé suspect ou malicieux
  - Exemple (et standard de facto) : Snort
- Détection par recherche d'anomalies
  - A partir d'un historique des activités réseau, on construit un profil type. Tout écart par rapport à ce profil est suspect.
  - Pas d'exemple véritablement opérationnel (SPADE/Snort)
- Détection des violations de politiques de sécurité
  - Qui a le droit de faire quoi, quand, comment, etc ? Quelles sont les actions prohibées ?
  - **Bro**

Architecture Bro 1.1



Architecture Bro 1.2



Libpcap

- Utilisée pour capturer et filtrer le trafic réseau.
- Bro Version: 1.2.1 (Fichier logs/info*)
- `Capture filter: ((((((((((((((((((((udp port 69) or (port ftp)) or (port 143)) or (tcp dst port 80 or tcp dst port 8080 or tcp dst port 8000)) or (tcp port 80 or tcp port 8080 or tcp port 8000 or tcp port 8001)) or (port 111)) or (tcp[13] & 7 != 0)) or (tcp src port 80 or tcp src port 8080 or tcp src port 8000)) or (port 6666)) or (port smtp)) or (icmp)) or (port 53)) or (port 111)) or ((ip[6:2] & 0x3fff != 0) and tcp)) or (port 6667)) or (port ftp)) or (port 512 or port 513 or port 515)) or (port telnet)) or (port smtp)) or (port 161 or port 162)) or (tcp[2:2] > 32770 and tcp[2:2] < 32901 and tcp[0:2] != 80 and tcp[0:2] != 22 and tcp[0:2] != 139)) or (port telnet or tcp port 513)) or (dst port 135 or dst port 137 or dst port 139 or dst port 445)`
- Comme tout utilitaire basé sur la libpcap, Bro peut relire un trafic capturé dans ce format.

Langage Bro

- Langage orienté Réseau, proche du C, exemple tftp.bro :

```
@load notice
redef enum Notice += { OutboundTFTP,   };
redef capture_filters += { ["tftp"] = "udp port
69" };
@load udp-common
@load site
global tftp_notice_count: table[addr] of count
&default = 0 &read_expire = 7 days;
event udp_request(u: connection)
{ if ( u$id$resp_p == 69/udp && u$id$orig_p >=
1024/udp )
{
   local src = u$id$orig_h;
   local dst = u$id$resp_h;
   if ( is_local_addr(src) && ! is_local_addr(dst) &&
     ++tftp_notice_count[src] == 1 )
   NOTICE([$note=OutboundTFTP, $conn=u,
   $msg=fmt("outbound TFTP: %s -> %s", src, dst)]);}
}
```

Event Engine

- Contrôle l'intégrité des paquets (en-têtes IP, etc), émet des alertes en cas d'anomalie et réassemblage des paquets fragmentés si nécessaire
- Si les tests d'intégrité sont corrects, les paquets sont inspectés dans leur contexte Session
  - IP source, IP destination, protocole, ports sources/destination
- TCP
  - Contrôles d'intégrité en-têtes et payload
  - Suivi de l'état (TCP flags) et mise à jour des tables de sessions
    - connection_attempt, connection_established, connection_rejected, connection_finished
- UDP
  - udp_request, udp_reply

Policy Script Interpreter

- Dernier étage de la fusée
- Traite les évènements (données en sortie de l'Event Engine) et applique les politiques décrites dans les scripts Bro.
- Ces scripts sont écrits dans le langage Bro.
  - Note : le langage Bro est également utilisé pour programmer des tâches d'administration internes comme la gestion des fichiers de log, notamment leur rotation.

Snort signatures not dead !

- Bro détecte les intrusions sur la base de politiques mais il est aussi possible d'intégrer les (bonnes vieilles) signatures Snort ou d'écrire les siennes.
- Exemple de signature Bro :
  - ```
    signature sid-1327 {
        ip-proto == tcp
        src-ip != local_nets
        dst-ip == local_nets
        dst-port == 22
        event "EXPLOIT ssh CRC32 overflow"
        tcp-state established,originator
        payload /\x00\x01\x57\x00\x00\x00\x18/
        payload /.{7}\xFF\xFF\xFF\xFF\x00\x00/
        }
    ```
    ```

Dynamic Protocol Detection

- Introduit dans la version 1.2 de Bro
- Objectif : ne plus baser l'analyse protocolaire sur les seuls ports destination.
  - Exemple (classique) : SSH sur port tcp/443
- Activation :
  - brolite.bro :
    - ## Dynamic Protocol Detection configuration
    - const use_dpd = T;
  - Le fichier dpd.sig contient les signatures des protocoles supportés. Exemple :

```
signature dpd_ssh_server {
    ip-proto == tcp
    payload /^[sS][sS][hH]-/
    tcp-state responder
}
```

Installation

- Physique :
  - Comme toute sonde réseau, une sonde Bro est rattachée à un point d'écoute du trafic.
- Logicielle :
  - Uniquement sur OS de la famille Linux/Unix
    - Linux, FreeBSD, Solaris, OpenBSD*
      - * Thanks to Jean-Philippe Luiggi
  - Version allégée
    - Bro-lite
      - N'installe que le minimum + configuration générique
    - Bro
      - Trilogie “configure + make + sudo make install”
    - Broccoli
      - Autorise les échanges entre sondes Bro dans un cluster Bro

Tour du propriétaire

- Arborescence Bro
  - Bin
  - Etc
    - bro.rc : script de démarrage/arrêt
    - bro.cfg : fichier de configuration “haut niveau”
      - Interfaces d'écoute
      - Valeur de certains “timers”
      - Emplacement des fichiers
      - Options de lancement du binaire bro
  - Logs
  - Policy : *.bro et répertoire sigs
  - Site : fichiers spécifiques

Chargement de la configuration

- bro.cfg
  - BRO_START_POLICY
    - @load site
      - Définition des paramètres réseau
    - @load brolite
      - “root policy”
        - Chargement de tous les scripts *.bro
        - Chargement des scripts de gestion des logs
          - rotate-logs.bro
        - Paramètres de détection des scans
        - Activation du DPD

Analyzers

- Génériques
  - type conn_id: record {
    orig_h: addr; # Address of originating host.
    orig_p: port; # Port used by originator.
    resp_h: addr; # Address of responding host.
    resp_p: port; # Port used by responder.
  };
  type endpoint: record {
    size: count; # Bytes sent by this endpoint so far.
    state: count; # The endpoint's current state.
  };
  type connection: record {
    id: conn_id;   # Originator/responder addresses/ports.
    orig: endpoint;   # Endpoint info for originator.
    resp: endpoint;   # Endpoint info for responder.
    start_time: time;   # When the connection began.
    duration: interval; # How long it was active (or has been so far).
    service: string;   # The service we associate with it (e.g., "http").
    addl: string;   # Additional information associated with it.
    hot: count;   # How many times we've marked it as
  sensitive.
  };

Analyzers

- Génériques (suite)
  - TCP
    - Etat : new_connection, connection_established, commection_attempt, connection_finished, connection_rejected, partial_connection, connection_half_finished, connection_partial_close, connection_pending
  - UDP
    - Etat : IP source/destination, port source/destination, timer
  - ICMP
  - “Hot” : traitement des spécificités de la politique de sécurité (IP autorisées ou non, protocoles interdits, etc.)
    - Détection spoofing par exemple
  - Scan : détection des scans

Analyzers

- Extraits du fichier logs/conn.*.log
  - Format
    - <start> <duration> <local IP> <remote IP> <service> <local port> <remote port> <protocol> <org bytes sent> <res bytes sent> <state> <flags> <tag>
    - 1167214427.838343 0.000073 xx.yy.94.187 aaa.bbb.62.202 http 54639 80 tcp ? ? REJ X
      1167214434.829760 0.000058 xx.yy.94.187 aaa.bbb.62.202 dns 38179 53 tcp ? ? REJ X
      1167214434.845444 0.000046 xx.yy.94.187 aaa.bbb.62.202 ssh 38812 22 tcp ? ? REJ X
    - 1167216096.519496 68.223067 aaa.bbb.62.202 xxx.yyy.2.191 ftp-data 32791 25052 tcp 0 4385674 SF X
    - 1167216094.844062 70.325961 aaa.bbb.62.202 xxx.yyy.2.191 ftp 32790 21 tcp 83 405 SF X #1 anonymous/-wget@

Analyzers

- Applicatifs
  - FTP, HTTP, SMTP, DNS, IRC, etc.
  - Exemples
    - FTP (logs/ftp.*.log)
    -
```
1167216094.844062 #1 1aaa.bbb.62.202/32790 >
131.243.2.191/ftp start
1167216095.259385 #1 response (220 ProFTPD 1.2.10
Server (FTPD) [131.243.2.191])
1167216095.259687 #1 USER anonymous/-wget@ (logged
in)
1167216095.687526 #1 SYST (215 UNIX Type: L8)
1167216095.895560 #1 PWD (done)
1167216096.103566 #1 TYPE I (ok)
1167216096.311520 #1 PASV (227
131.243.2.191/25052)
1167216096.727545 #1 RETR bro-1.1d-stable.tar.gz
(complete)
1167216165.170067 #1 finish
```

Analyzers

- FTP (suite)
  - 1168894305.269184 #1 aaa.bbb.32.85/4453 > xxx.yyy.171.25/**21124** start
  - 1168894305.711578 #1 response (220 fhjjgff wertyhfg gju..)
    1168894305.711578 #1 USER 1 (logged in)
    1168894305.856331 #1 PORT aaa,bbb,32,85,26,225 (ok)
    1168894309.279371 #1 ftp-data <unknown> '<unknown>'
    1168894306.051945 #1 RETR ip_64312.exe (complete)
    1168894317.073481 #1 QUIT (closed)
    1168894319.110916 #1 finish
    1168898131.270282 #2 aaa.bbb.32.85/1459 > 166.104.216.37/2755 start
    1168898132.784567 #2 response (220 .)
    1168898132.784567 #2 USER 1 (logged in)
    1168898133.285277 #2 PORT aaa,bbb,32,85,66,252 (ok)
    1168898134.304389 #2 ftp-data <unknown> '<unknown>'
    1168898133.796701 #2 RETR 2.exe (complete)
    1168898134.848643 #2 QUIT (closed)
    1168898135.996536 #2 finish
  -

Analyzers

- Applicatifs (suite)
  - HTTP
    - 1167731797.392553 %5 GET /Repair Registry Pro.exe (200 "OK" [343625] www.regfixit.com)
  - IRC
    - 1167747254.149324 #1 new connection aaa.bbb.208.235/59476 > xxx.yyy.32.85/IRC
    - 1167747254.149324 #1 new user, user='nepenthes', host='gambetta', server='xxx.yyy.32.85', real = 'gaim'
    - 1167747254.247123 #1 changing nick name to 'toto'
    - 1167747422.037966 #1 user '' joined '#mychannel'
    - 1167747422.050551 #1 user 'toto' joined '#mychannel'

Analyzers

- IRC – Botnets (*) :
  - 1167544853.786941 #3 new connection ppp.rrr.50.194/2661 > zzz.uuu.95.1/**8000**
  - 1167544853.786941 #3 changing nick name to 'sawu'
  - 1167544853.786941 #3 new user, user='sefu', host='.', server='.', real= '3dacanu piticu 9... 3atunci cine 7?MODES +isx '
  - 1167544853.786941 #3 changing nick name failed
  - 1167544853.787655 #3 changing nick name to 'sawu_'
  - 1167544853.953961 #3 changing nick name failed
  - 1167544856.754177 #2 user '' joined '#hatshells'
    - Source: J.P Luiggi

Conclusion

- Un NIDS très très prometteur
  - NIDS est peut-être un terme malheureux : Bro est plus un (excellent) analyseur protocolaire,
- Peu de faux positifs
  - La gestion des connexions/sessions fonctionne bien.
  - L'analyse protocolaire est un réel atoût.
- Mais
  - Demande un investissement en temps non négligeable si l'on souhaite :
    - “customiser” les scripts de politique
      - Apprentissage du langage Bro
    - Comprendre et maitriser le fonctionnement du moteur Bro
  - Manque d'une interface graphique / SGBD native
    - Sauf pour les inconditionnels du fichier Texte
- Peut efficacement compléter une panoplie d'IDS

Merci

Template OpenOffice : Chih-Hao Tsai, http://technology.chtsai.org/impress/