



IpMorph :

« vers la mystification de la prise d'empreinte de pile »



Présentation à l'OSSIRB
17 février 2009

<http://dev.ipmorph.org>



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Sommaire

- **Contexte**
- **Historique**
- **Etymologie**
- **Techniques de détection**
- **Etat de l'art de la mystification**
- **Utilisation d'IpMorph**
- **IpMorph != Packet Purgatory / Morph**
- **Socle logiciel**
- **Architecture générale**
- **Translation ARP**
- **Mystification de la congestion**
- **Focus : Calcul ISN**
- **Focus : Interfaces réseaux « read/write »**
- **Personality Manager**
- **Démonstration**
- **Etat d'avancement**
- **Perspectives**
- **Bibliographie**

IpMorph : « vers la mystification de la prise d'empreinte de pile »



Contexte

Théorème :

« Vivons heureux, vivons caché »

Corolaire :

« Si une machine peut falsifier son identité au yeux des outils de reconnaissance et usurper celle d'un système moins intéressant pour la menace, celle ci minimise l'attrait de l'attaquant et perturbe la pertinence des attaques ciblées à sa nature apparente. »

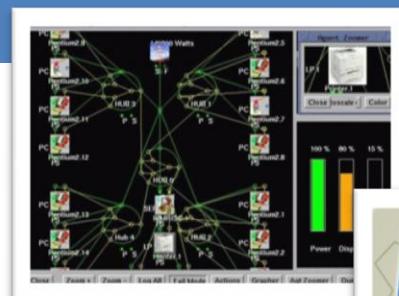




IpMorph : « vers la mystification de la prise d'empreinte de pile »

Historique

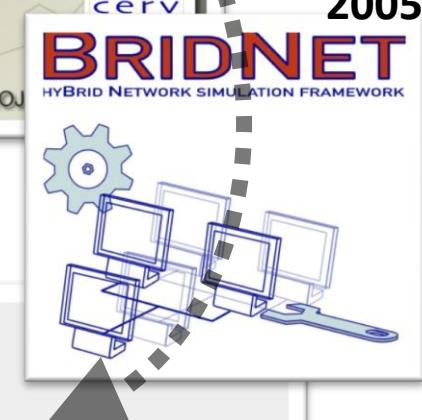
- **GFOX-0 & GFOX-1**
 - Simulation multi-agents de la confidentialité, de l'intégrité et de la disponibilité d'un système d'information
- **EREBOR**
 - Vers l'hybride « réel/virtuel »
- **BridNet**
 - Première pile « UserLand TCP/IP »
- **Hynesim**
 - Plateforme de virtualisation de système d'information



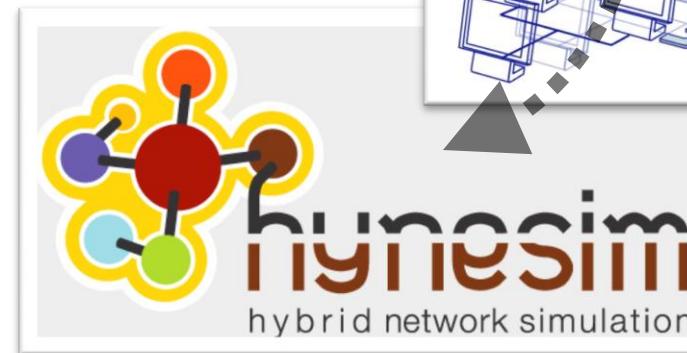
1999



2002



2005



2008-?



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Etymologie

– IpMorph

- Du néo-grec numérique μορφή - θημωνιά, *morphéstakis*, « forme de pile ».

– Suffixe

- -morph /mɔʁf/

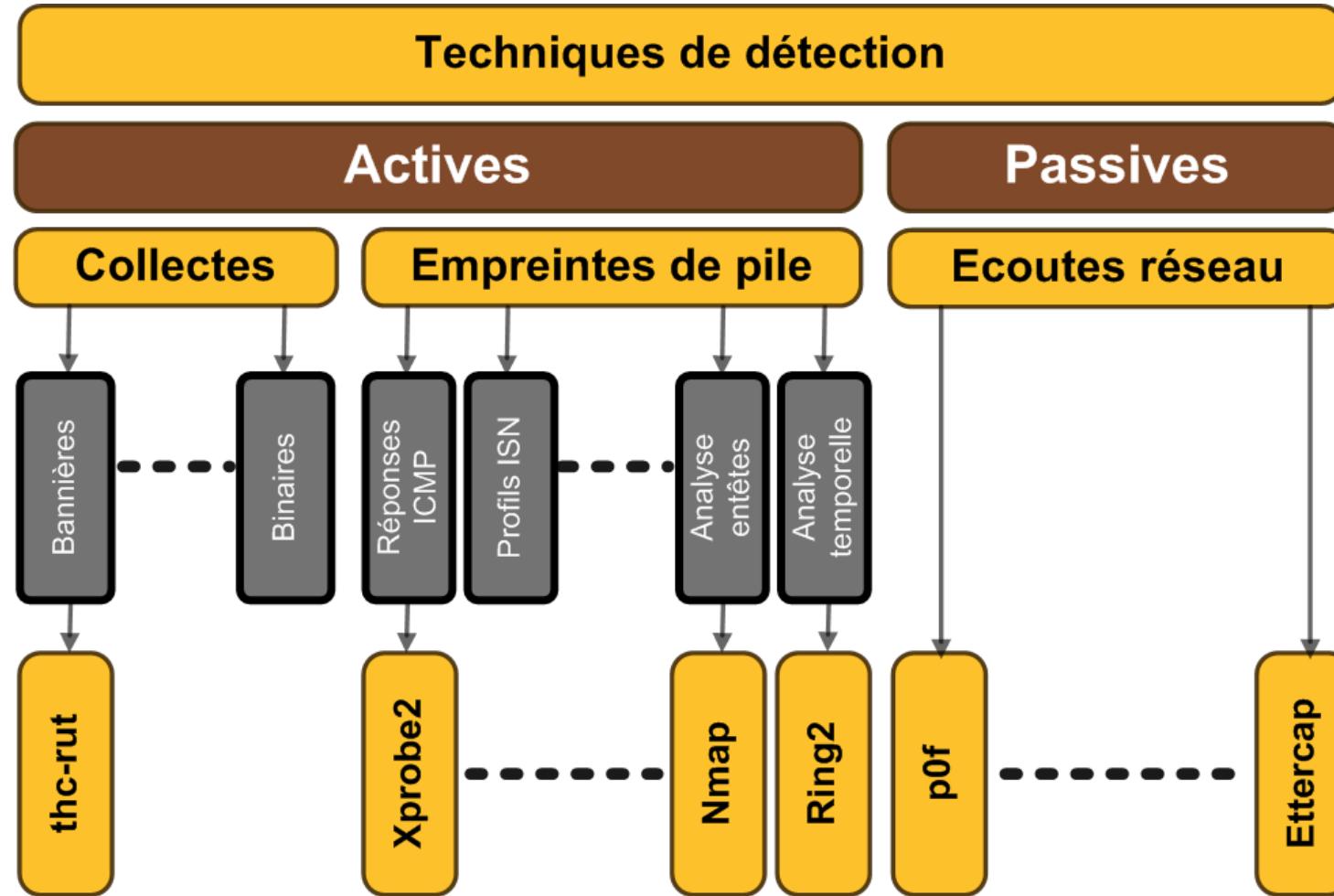
1. En relation avec la forme d'une pile IP, qui a la forme d'une pile IP.

– Apparentés étymologiques

1. FpFucker
2. OS fingerprint frustrating

IpMorph : « vers la mystification de la prise d'empreinte de pile »

Techniques de détection [11]





IpMorph : « vers la mystification de la prise d'empreinte de pile »

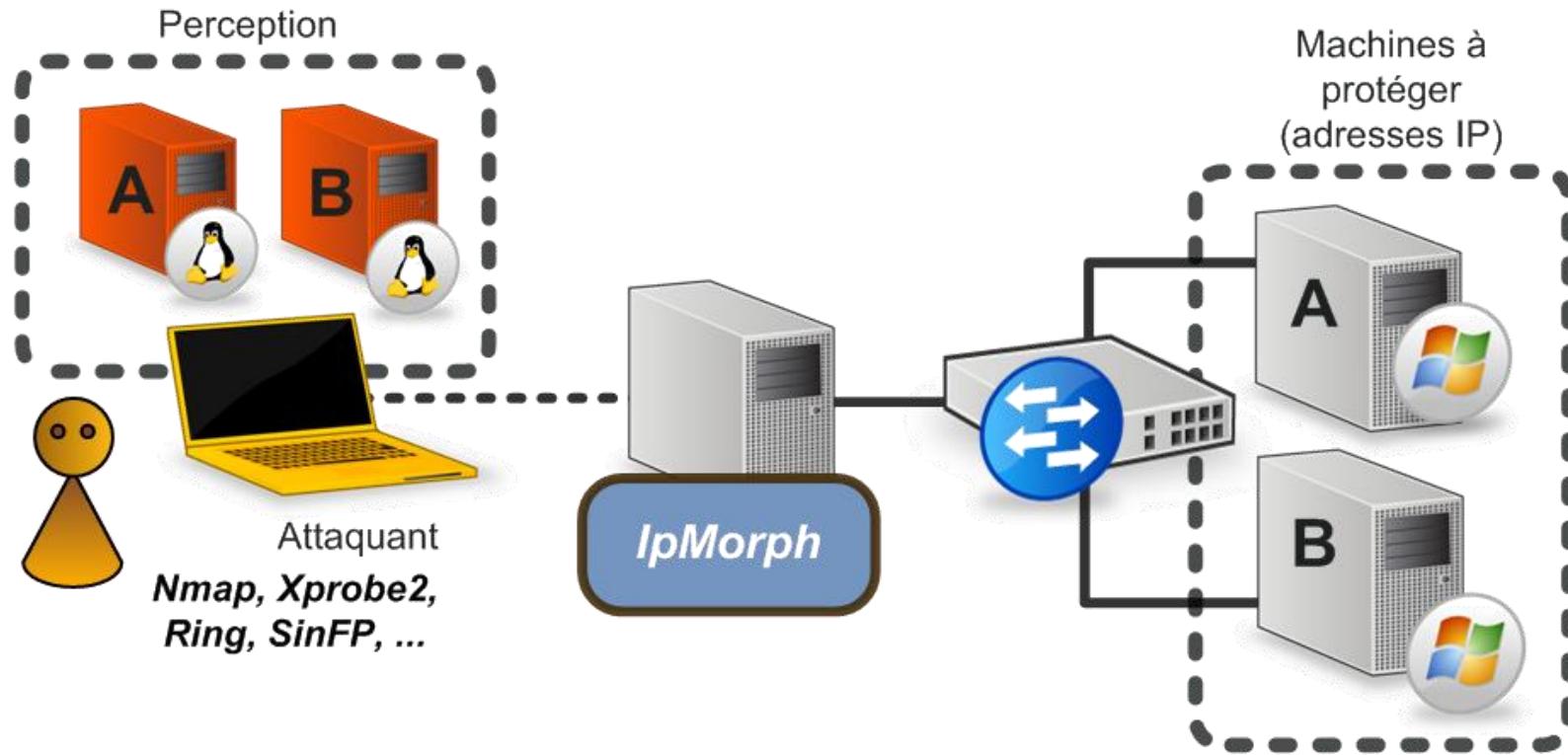
Etat de l'art de la mystification [7]

- **Filtrage**
 - Stealth patch : **Unmaintained as of 2002, GNU/Linux kernel 2.2-2.4** [14]
 - Blackhole : **FreeBSD, kernel options** [16]
 - IPlog : **Unmantained as of 2001, *BSD** [17]
 - Packet filter : **OpenBSD** [18]
- **Configuration et modification de pile TCP/IP ("host based")**
 - Ip Personality [19]
 - Fingerprint Fucker [12][13]
 - Fingerprint scrubber [1]
 - OSfuscate [8]
- **Substitution de pile TCP/IP ("proxy behaviour")**
 - Honeyd [9]
 - Packet purgatory / Morph [10]

IpMorph : « vers la mystification de la prise d'empreinte de pile »



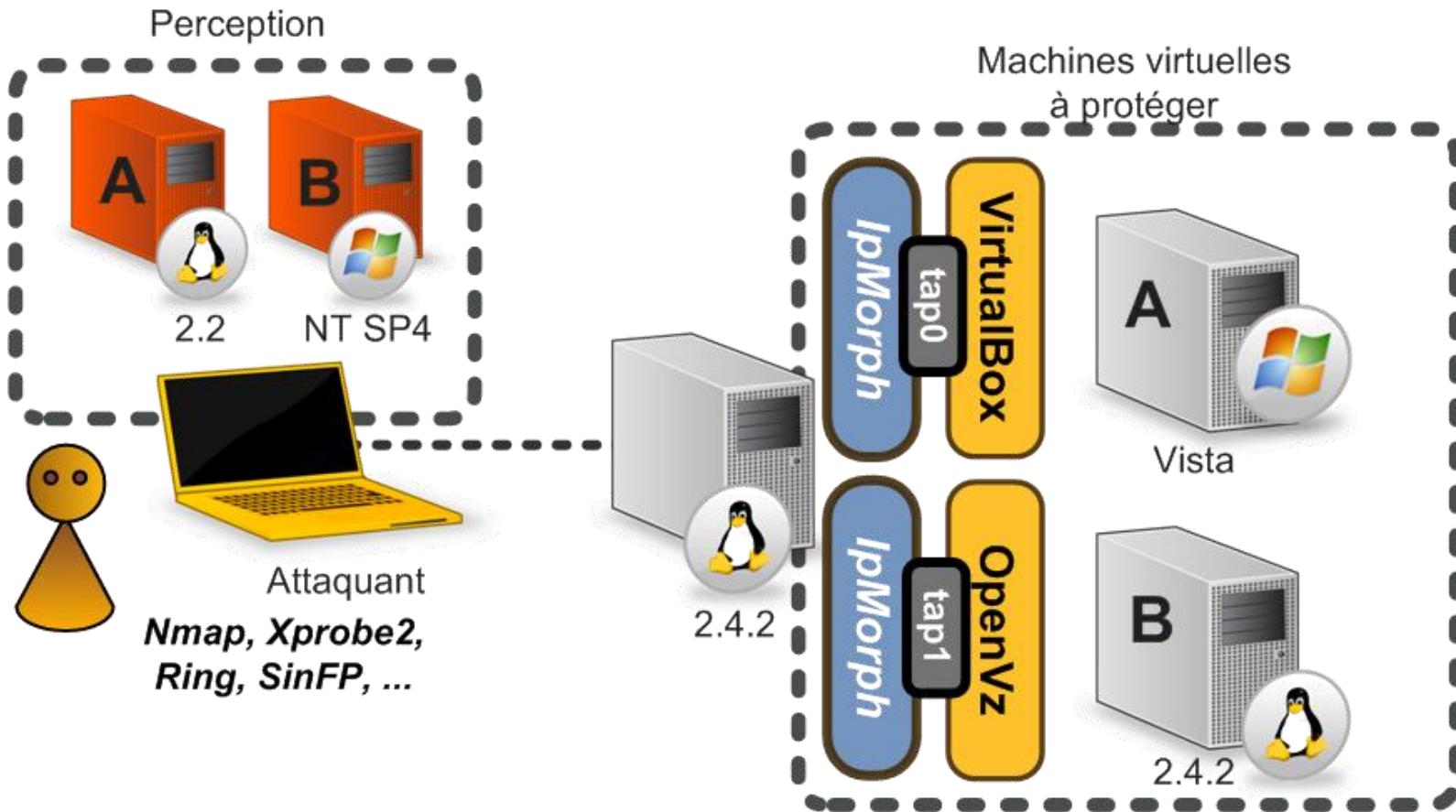
Utilisation d'IpMorph 1/2



IpMorph en coupure de machines réelles

IpMorph : « vers la mystification de la prise d'empreinte de pile »

Utilisation d'IpMorph 2/2



IpMorph en relai protection de machines virtuelles

IpMorph : « vers la mystification de la prise d'empreinte de pile »

IpMorph != Packet Purgatory / Morph [20]



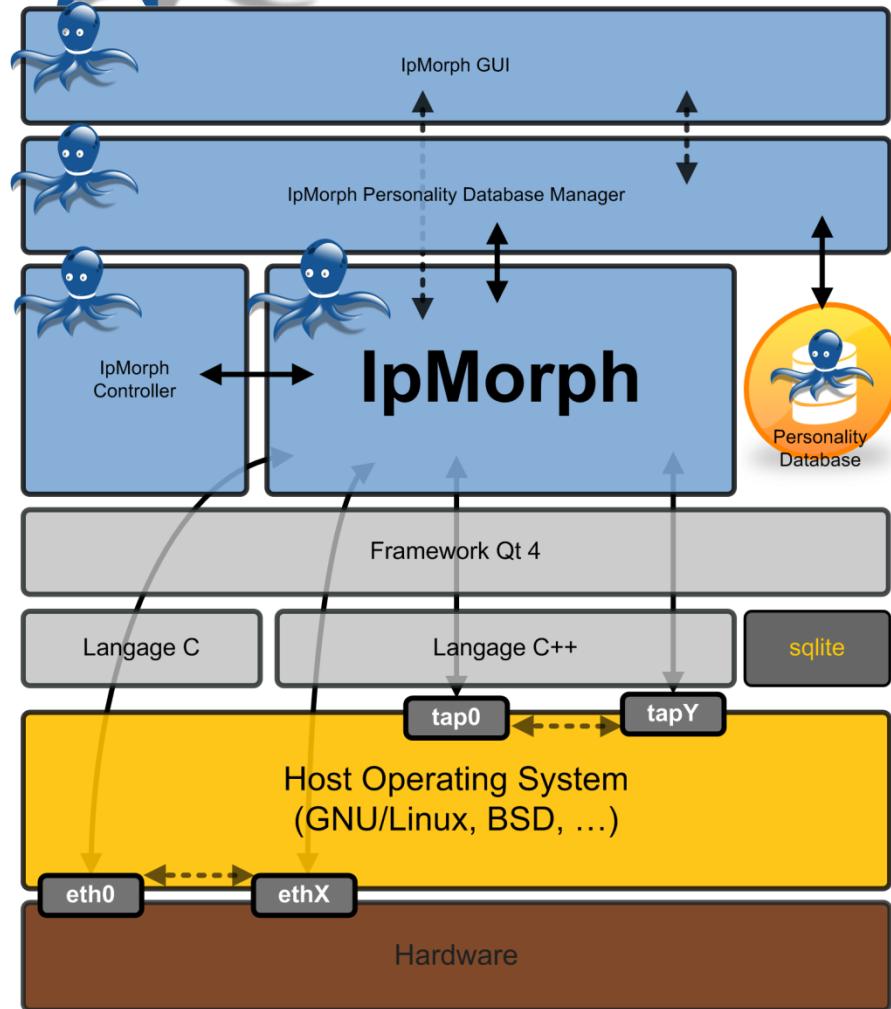
A screenshot of a Gedit text editor window showing C code for an 'outbound.c' file. The code handles different operating systems based on the 'os_type' field in a 'morph_handle' structure. Three specific cases are highlighted with arrows pointing to callouts:

- Microsoft Windows 2000**: Case for OS_TYPE_WIN2K. The code sets the IP ID to the current value plus one, sets the IP TTL to 128, and sets the TCP THRESHOLD to 0x402E.
- OpenBSD**: Case for OS_TYPE_OPENBSD_33. The code generates a random IP ID, sets the IP TTL to 255, and sets the TCP THRESHOLD to 0x403D.
- GNU/Linux 2.4**: Case for OS_TYPE_LINUX_24. The code sets the IP ID to 0, sets the IP TTL to 64, and sets the TCP THRESHOLD to 0x40DE.

```
outbound.c (~/Devel/morph-0.3) - gedit
Eichier Édition Affichage Rechercher Outils Documents Aide
Nouveau Ouvrir Enregistrer Imprimer... Annuler Rétablir Couper Copier Coller Rechercher Remplacer
Documents
outbound.c
111 tmp_loc = (uint8_t *)(((uint8_t *)tcp_header)+(tcp_header->th_off*4));
112
113 switch(morph_handle->os_type) {
114     case OS_TYPE_WIN2K:
115         ip_header->ip_id=htons(morph_handle->ip_id++);
116         ip_header->ip_off=htons(IP_DF);
117         ip_header->ip_ttl=128;
118         tcp_header->th_win = htons(0x402E);
119
120
121         if(tcp_header->th_flags == (TH_SYN|TH_ACK)) {
122             tcp_option_reorder(morph_handle, packet);
123         }
124
125         break;
126
127     case OS_TYPE_OPENBSD_33:
128         ip_header->ip_id=rand_uint16(morph_handle->randomizer),
129         ip_header->ip_ttl=255;
130         tcp_header->th_win = htons(0x403D);
131         ip_header->ip_off=htons(IP_DF);
132
133
134         if(tcp_header->th_flags == (TH_SYN|TH_ACK)) {
135             tcp_option_reorder(morph_handle, packet);
136         }
137         break;
138
139     case OS_TYPE_LINUX_24:
140         ip_header->ip_id=0;
141         ip_header->ip_ttl=64;
142         ip_header->ip_off=htons(IP_DF);
```

IpMorph : « vers la mystification de la prise d'empreinte de pile »

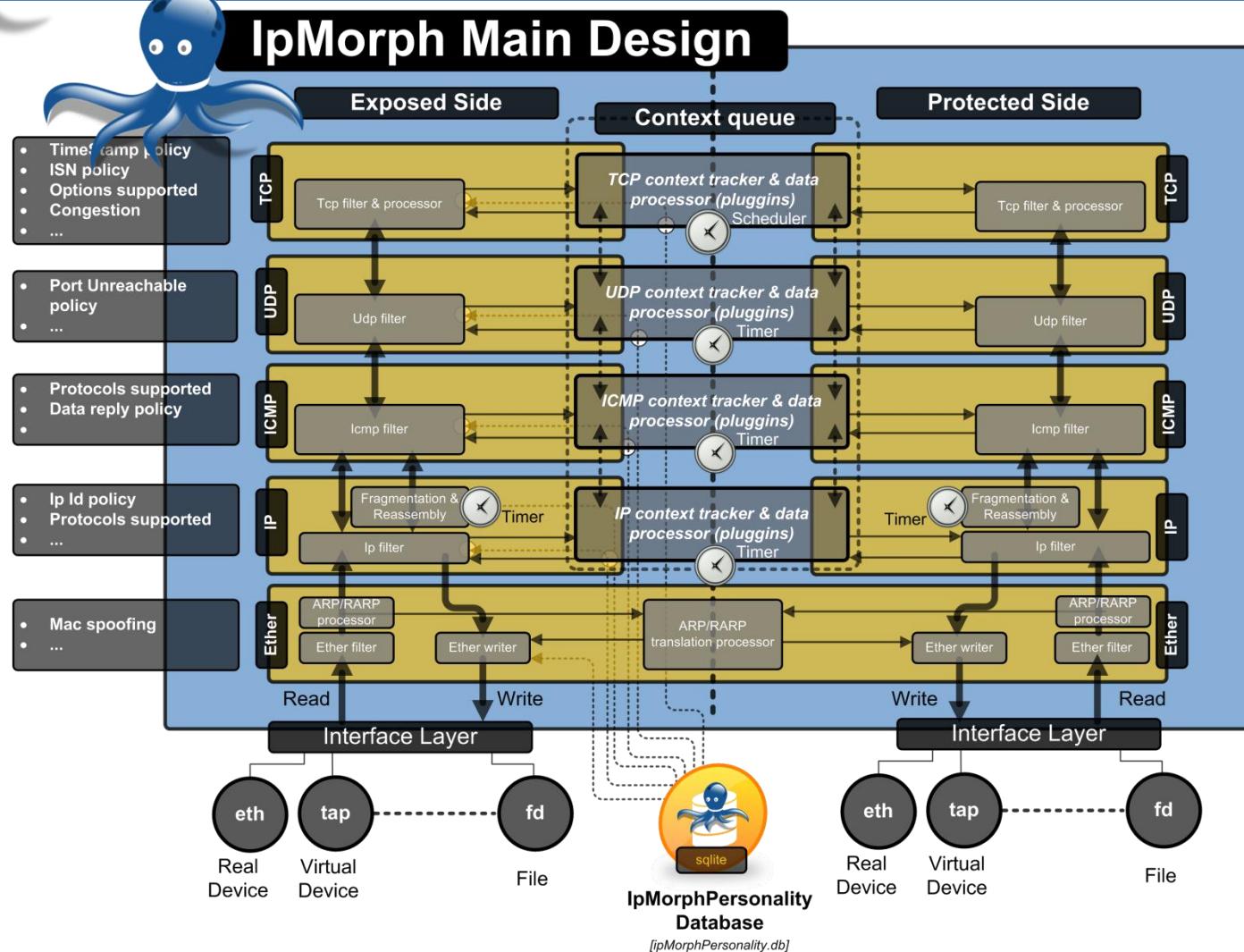
Socle logiciel



- **Langage C++**
- **Application « UserLand »**
- **Framework Qt4**
- **Eléments constituants :**
 - IpMorph (Core)
 - IpMorph Controller
 - IpMorph Personality Manager
 - IpMorph Personality Database
 - IpView (IpMorph GUI)
- **Portabilité :**
 - GNU/Linux
 - *BSD, Mac OS
- **License GPLv3**

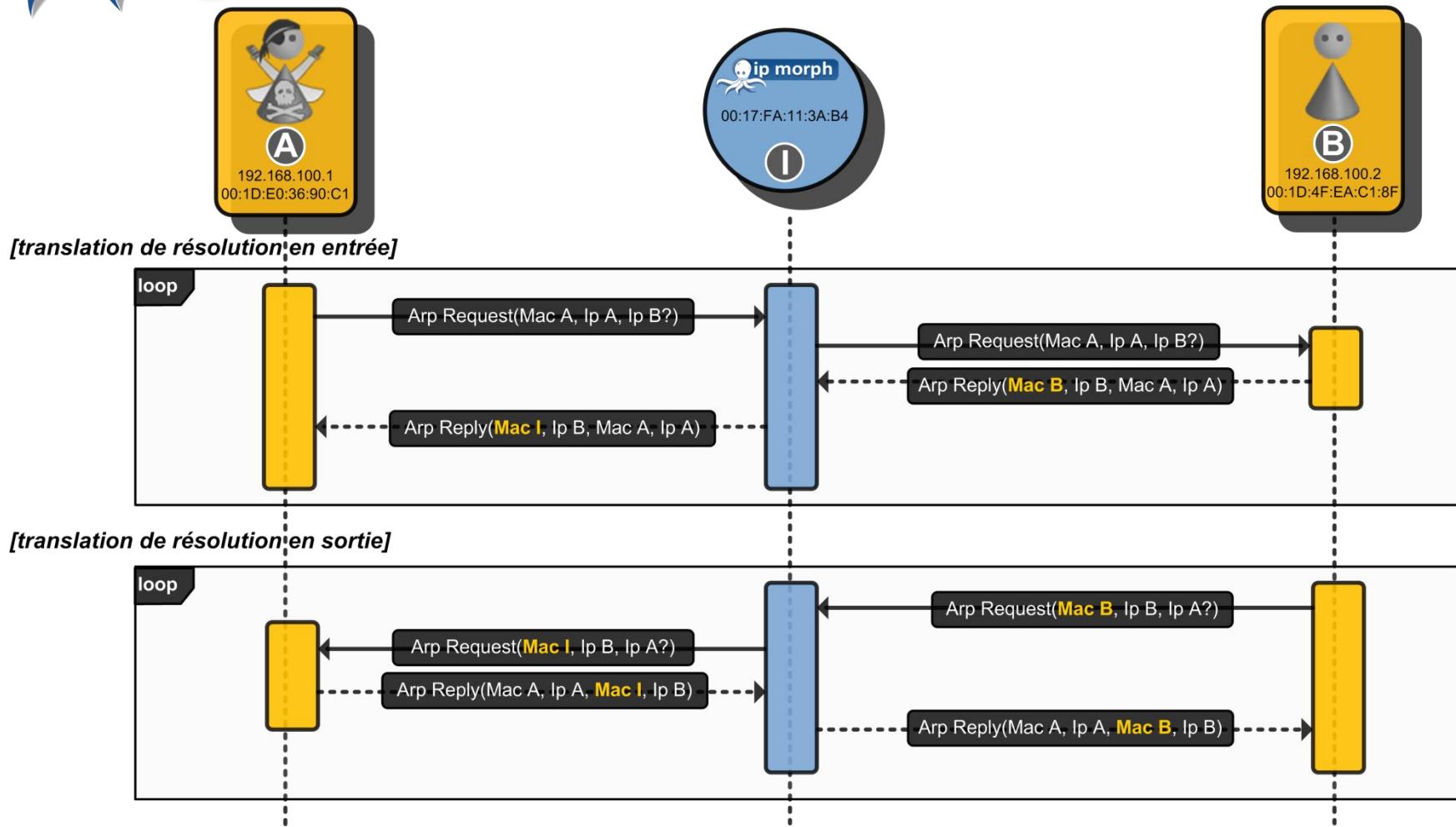
IpMorph : « vers la mystification de la prise d'empreinte de pile »

Architecture générale



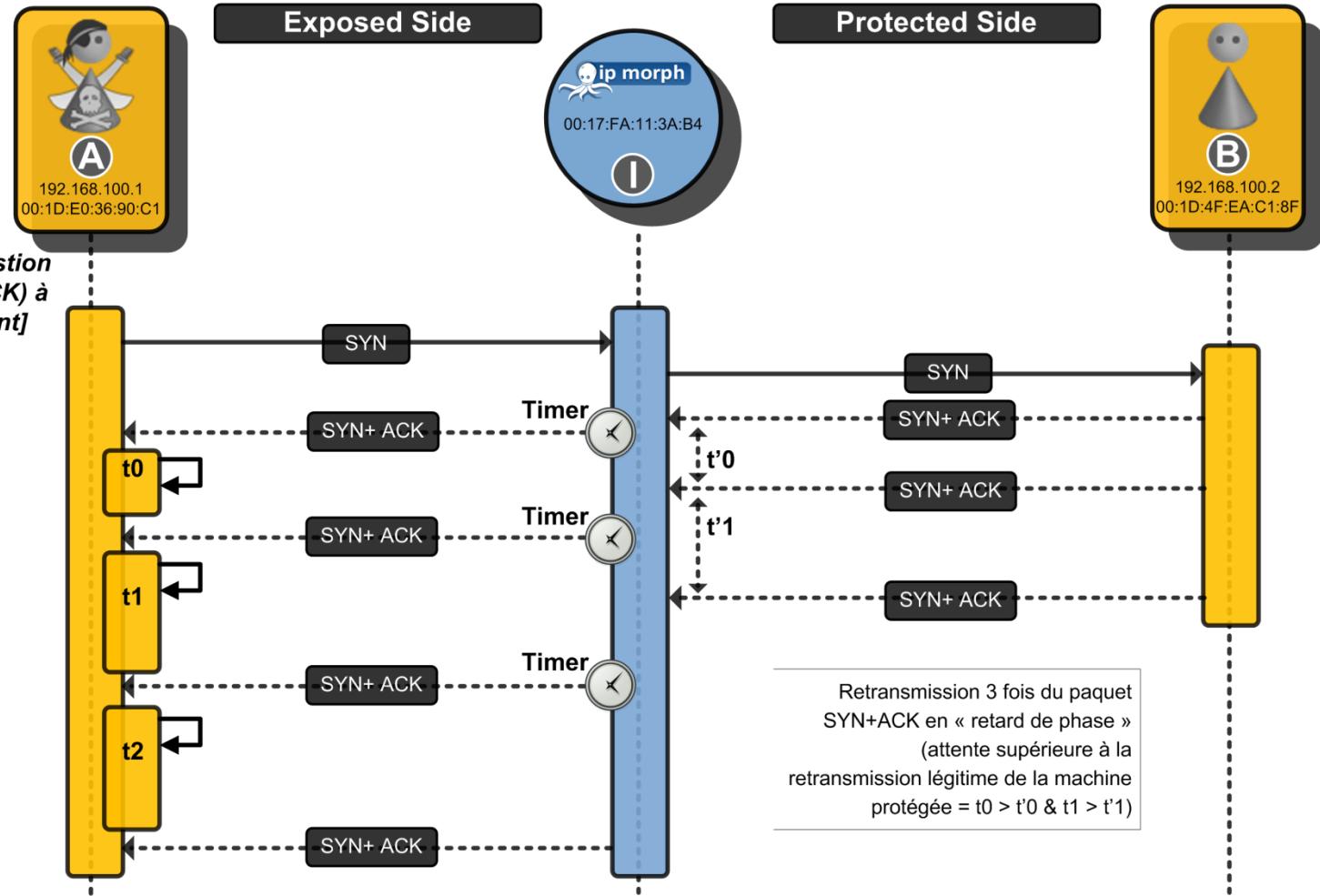
IpMorph : « vers la mystification de la prise d'empreinte de pile »

Translation ARP



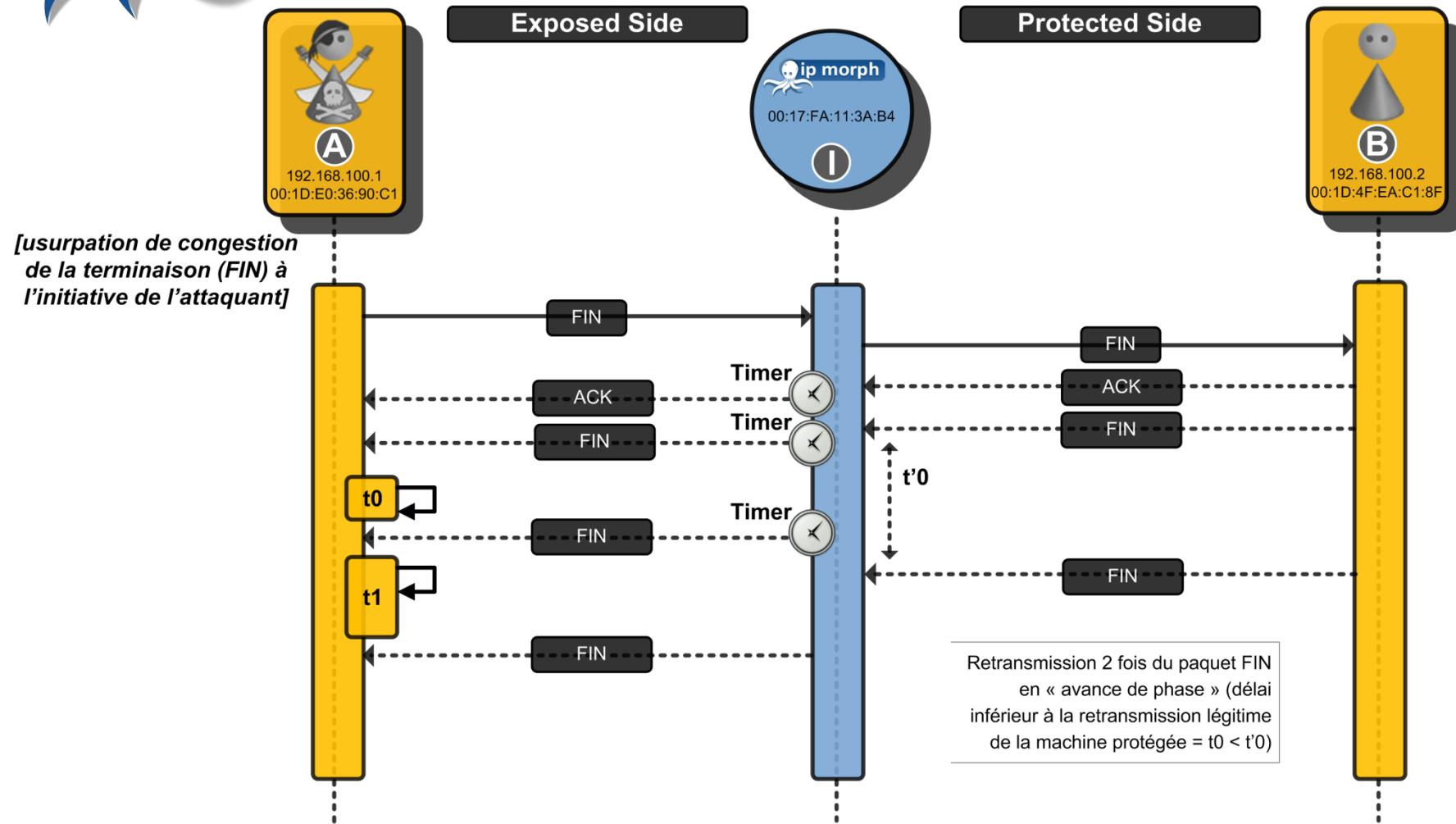
IpMorph : « vers la mystification de la prise d'empreinte de pile »

Mystification de la congestion - SYN+ACK



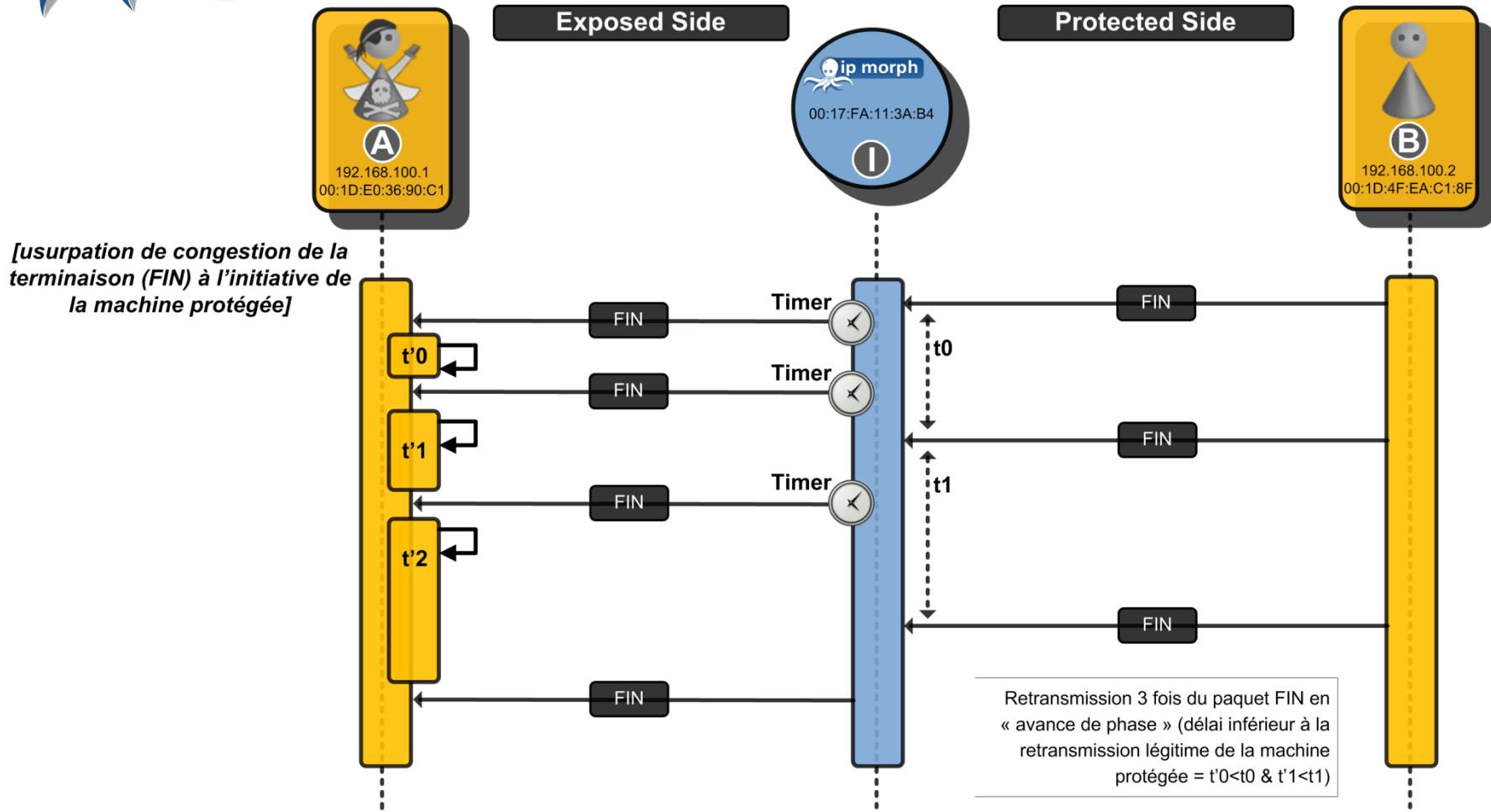
IpMorph : « vers la mystification de la prise d'empreinte de pile »

Mystification de la congestion - FIN+ACK « client »



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Mystification de la congestion - FIN+ACK « serveur »



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Focus : Calcul ISN



ipMorphTcpContext.cpp (~) - gedit

Nouveau Ouvrir Enregistrer Imprimer... Annuler Rétablir Couper Copier Coller Rechercher Remplacer

```

1209 if(_NOS.getTcpIsnMode() == NmapOS::ISN_TR) // True Random
1210 {
1211     _isn = rand();
1212 }
1213 else if(_NOS.getTcpIsnMode() == NmapOS::ISN_RI) // Random Increment
1214 {
1215     // méthode de Box-Muller
1216     // r et phi aléatoires entre 0 et 1 : (0 1]
1217     // calcul des nombres de la loi normale :
1218     // z0 = cos(2*PI*phi) * sqrt(-2*ln(r))
1219     // z1 = sin(2*PI*phi) * sqrt(-2*ln(r))
1220     // et on obtient nos 2 nombres avec le bon écart-type autour de la moyenne :
1221     // n0 = moyenne + ecart-type * z0
1222     // n1 = moyenne + ecart-type * z1
1223     // note : en C, ln() correspond à la fonction "log()" de math.h
1224
1225     unsigned long ecartType = _NOS.getTcpIsnParm();
1226     unsigned long moyenne = 2*ecartType;
1227
1228     srand(_getTime() + _isn); // réinitialisation de la génération des nombres aléatoires
1229     double r = ((double)1*(1+rand())/(RAND_MAX+1.0));
1230
1231     srand(_getTime() + (int)(_isn * (1+r))); // réinitialisation de la génération des nombres aléatoires
1232     double phi = ((double)1*(1+rand())/(RAND_MAX+1.0));
1233
1234     double z0 = cos(2*M_PI*phi) * sqrt(-2*log(r));
1235     double n0 = moyenne + ecartType * z0;
1236
1237     _isn = (long)n0;
1238 }
1239 else if(_NOS.getTcpIsnMode() == NmapOS::ISN_TD) // Time Dependant
1240 {
1241     unsigned int gcd = _NOS.getTcpIsnParm();
1242     if(gcd!=0)
1243     {
1244         _isn+=gcd;
1245     }
1246     else
1247     {
1248         unsigned int avg = 10;
1249         _isn+=avg+lrint((double)avg*rand()/(RAND_MAX+1.0));
1250     }
1251 }
1252 else if(_NOS.getTcpIsnMode() == NmapOS::ISN_64K) // 64K
1253 {
1254     _isn+=64000;
1255 }
1256 else if(_NOS.getTcpIsnMode() == NmapOS::ISN_i800) // i800
1257 {
1258 }
```

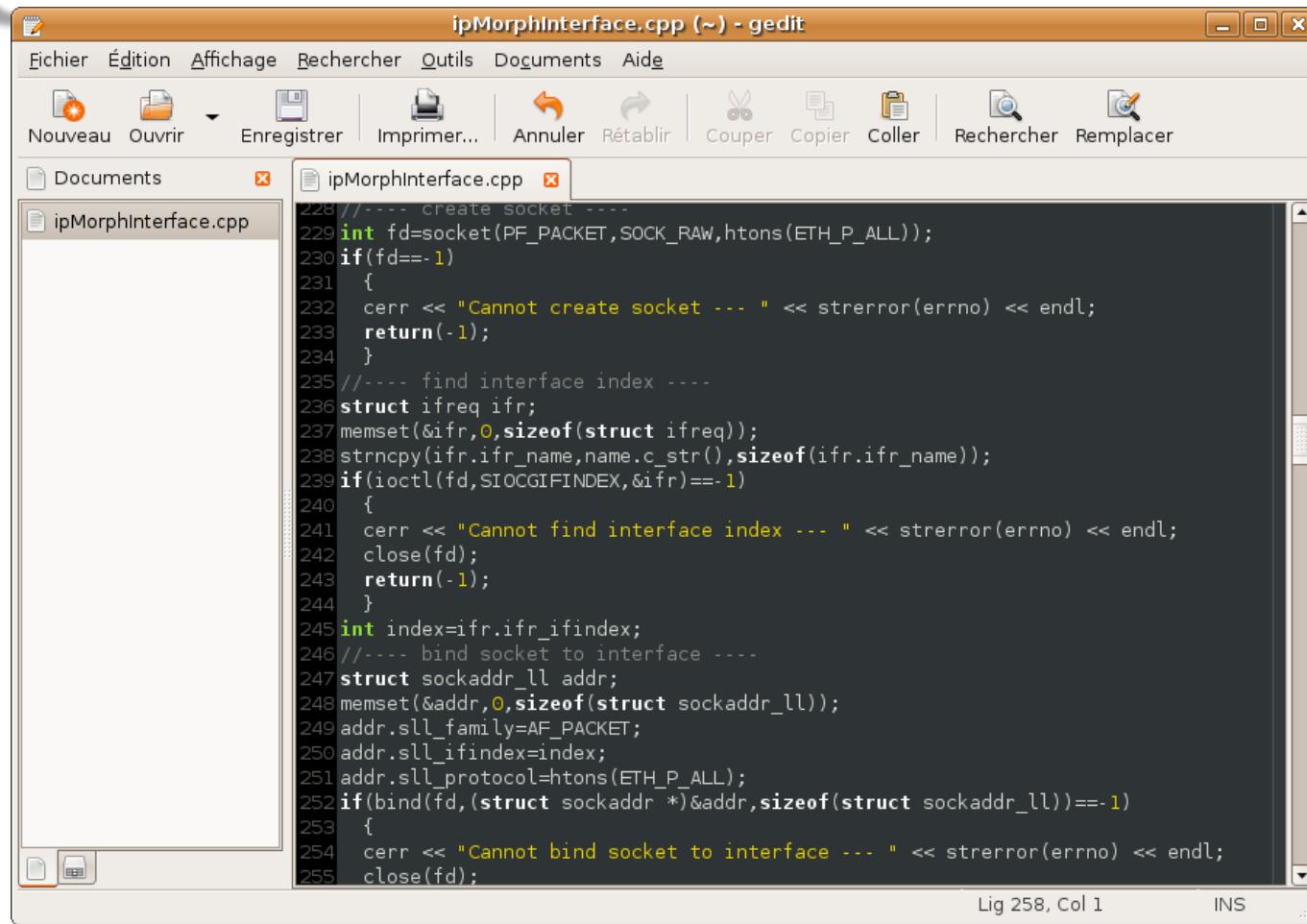
Lig 1235, Col 1 INS

Génération de l'ISN pour l'ancienne version de Nmap (<4.11)



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Focus : Interfaces réseaux « read/write »



The screenshot shows a Gedit window titled "ipMorphInterface.cpp (~) - gedit". The code in the editor is as follows:

```
228 //---- create socket ----
229 int fd=socket(PF_PACKET,SOCK_RAW,htons(ETH_P_ALL));
230 if(fd== -1)
231 {
232     cerr << "Cannot create socket --- " << strerror(errno) << endl;
233     return(-1);
234 }
235 //---- find interface index ----
236 struct ifreq ifr;
237 memset(&ifr,0,sizeof(struct ifreq));
238 strncpy(ifr.ifr_name,name.c_str(),sizeof(ifr.ifr_name));
239 if(ioctl(fd,SIOCGIFINDEX,&ifr)== -1)
240 {
241     cerr << "Cannot find interface index --- " << strerror(errno) << endl;
242     close(fd);
243     return(-1);
244 }
245 int index=ifr.ifr_ifindex;
246 //---- bind socket to interface ----
247 struct sockaddr_ll addr;
248 memset(&addr,0,sizeof(struct sockaddr_ll));
249 addr.sll_family=AF_PACKET;
250 addr.sll_ifindex=index;
251 addr.sll_protocol=htons(ETH_P_ALL);
252 if(bind(fd,(struct sockaddr *)&addr,sizeof(struct sockaddr_ll))== -1)
253 {
254     cerr << "Cannot bind socket to interface --- " << strerror(errno) << endl;
255     close(fd);
}
```

The status bar at the bottom of the editor indicates "Lig 258, Col 1" and "INS".

Ouvertures des interfaces (eth, tap) pour capture et injection



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Personality Manager : Structure de l'index

Screenshot of a Windows-based XML editor showing the file `ipMorphFpIndex.xml`. The code is an XML document defining personality categories and entries for OS fingerprinting.

```

<description>This category contains all Microsoft OSes</description>
<keywords></keywords>
<ipmPersonalityCategories>
  <ipmPersonalityCategory shortName="Win NT" longName="">
    <keywords></keywords>
    <longname><![CDATA[Windows NT family]]></longname>
    <description><![CDATA[Windows NT, 2000, XP, .NET]]></description>
    <ipmPersonalityEntries>
      <ipmPersonalityEntry shortName="Win XP" longName="">
        <description/>
        <ipmToolsSignatures>
          <ipmToolsSignature toolId="nmap2"><![CDATA[Microsoft Windows XP]]></fingerprint>
          <ipmToolsSignature toolId="xprobe2"><![CDATA[Microsoft Windows XP]]></fingerprint>
          <ipmToolsSignature toolId="sinfp"><![CDATA[80]]></fingerprint>
        <ipmToolsSignatures>
        <ipmPersonalityEntry shortName="Win 2000 SP4" longName="Windows 2000 Workstation with Service Pack 4">
          <description/>
          <ipmToolsSignatures>
            <ipmToolsSignature toolId="nmap2"><![CDATA[Microsoft Windows 2000 SP4]]></fingerprint>
            <ipmToolsSignature toolId="xprobe2"><![CDATA[Microsoft Windows 2000 Workstation SP4]]></fingerprint>
            <ipmToolsSignature toolId="sinfp"><![CDATA[44]]></fingerprint>
            <ipmToolsSignature toolId="ring2"><![CDATA[Windows 2K RINGv2]]></fingerprint>
            <ipmToolsSignature toolId="satory"><![CDATA[Windows 2000]]></ipmToolsSignature>
            <ipmToolsSignature toolId="p0f"><![CDATA[Windows:2000 SP4]]></ipmToolsSignature>
          <ipmToolsSignatures>
          <ipmPersonalityEntry shortName="Win NT 4.0 Workstation" longName="">
            <description/>
            <ipmToolsSignatures>
              <ipmToolsSignature toolId="nmap2"><![CDATA[Microsoft Windows NT 4.0 Workstation SP6a]]></fingerprint>
              <ipmToolsSignature toolId="xprobe2"><![CDATA[Microsoft Windows NT 4 Workstation Service Pack 6a]]></fingerprint>
              <ipmToolsSignature toolId="sinfp"><![CDATA[47]]></fingerprint>
              <ipmToolsSignature toolId="ring2"><![CDATA[Windows NT4 (or 98) RINGv2]]></fingerprint>
              <ipmToolsSignature toolId="satory"><![CDATA[Windows NT 4]]></ipmToolsSignature>
              <ipmToolsSignature toolId="p0f"><![CDATA[Windows:NT 4.0 SP6a (1)]]></ipmToolsSignature>
            <ipmToolsSignatures>
          </ipmPersonalityEntries>
        </ipmPersonalityCategory>
      </ipmPersonalityEntries>
    </ipmPersonalityCategory>
  </ipmPersonalityCategories>
</ipmFpIndex>

```

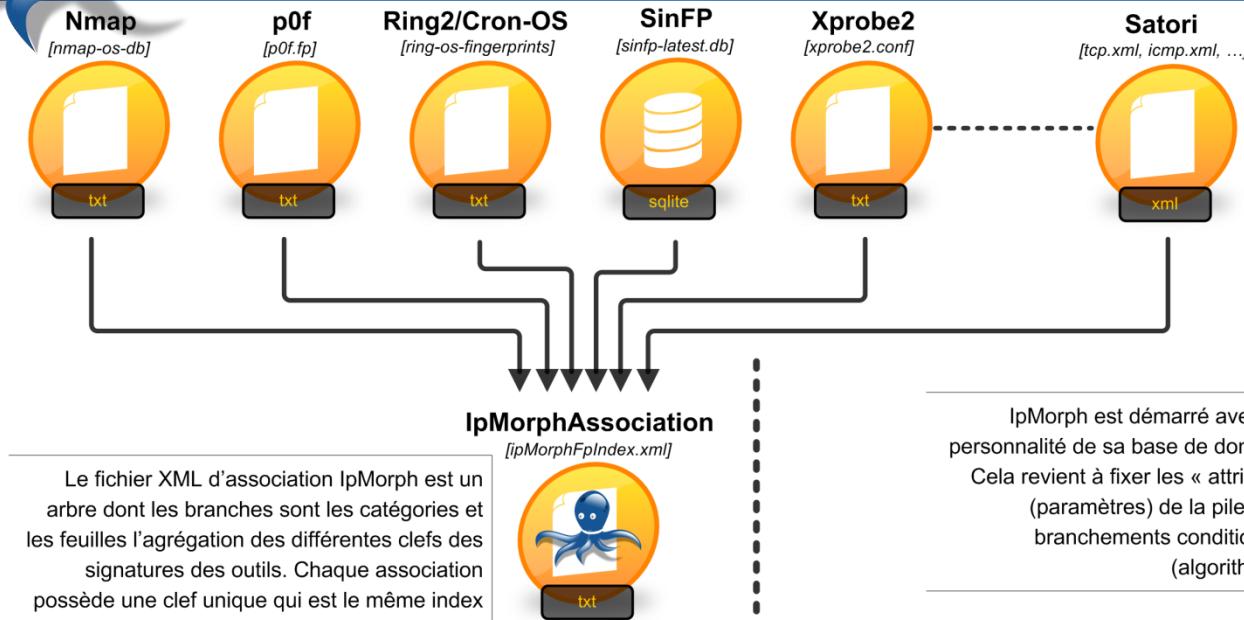
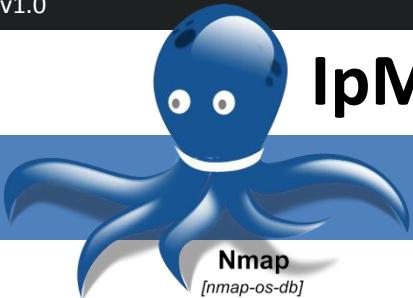
Annotations in the screenshot:

- A callout box labeled "Nmap, Xprobe2 & SinFP" points to the first set of signatures (lines 16-21).
- A callout box labeled "Nmap, Xprobe2, SinFP, RINGv2, Satory & p0f" points to the second set of signatures (lines 24-39).

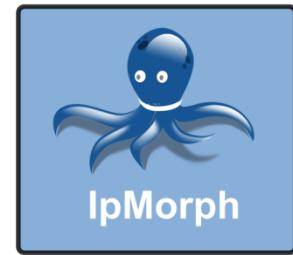
Formalisation XML et association des signatures des outils d'OSFP

IpMorph : « vers la mystification de la prise d'empreinte de pile »

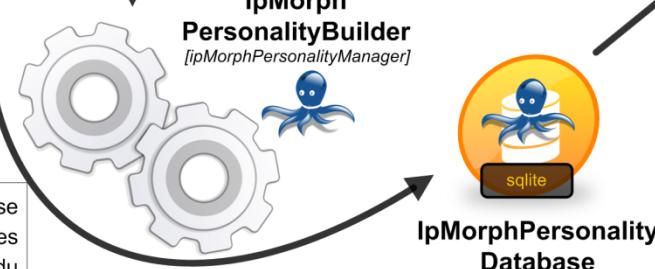
Personality Manager : Présentation



IpMorph est démarré avec une personnalité de sa base de données. Cela revient à fixer les « attributs » (paramètres) de la pile et les branchements conditionnels (algorithmes).



Le processus de construction (ou de régénération) de la base de données des personnalités effectue la fusion des paramètres des signatures des différents outils en fonction du fichier XML d'association et génère une personnalité IpMorph pour chaque noué *</pmAssoc>*. Un contrôle d'incohérence est effectué et la résolution des conflits nécessite l'intervention de l'utilisateur.



`loadPersonality(UniqueId)`

IpMorph : « vers la mystification de la prise d'empreinte de pile »

Personality Manager : Démonstration

IpMorph Personality Manager GUI v.0.1

File Action Window Help

Title

p0f.fp

- OpenBSD
 - OpenBSD 3.0-3.4 (Opera?)
- Windows
 - Windows XP/2000 while downloading (leak)
- p0f
 - p0f sendsyn utility
- @Checkpoint
 - @Checkpoint (unknown 1)
 - @Checkpoint (unknown 2)
- @Eagle
 - @Eagle Secure Gateway
- @Mysterious
 - @Mysterious Chello proxy (?)
 - @Mysterious GPRS gate
- @Windows
 - @Windows 98
 - @Windows XP/2000
 - @Windows XP/2000 (RF)
 - tcp_options = M1460,E
 - @Windows XP/2000 (RF)
 - quirks = P
 - @Windows XP/2000 (RF)
 - system_class = @Mysterious
 - @Windows XP/2000 (RF)
 - os_name = Chello proxy (?)
 - @Windows XP/2000 (RF)
 - CT1323+, W-, timestamp+)
 - AIX
 - AIX 4.3
 - AIX 4.3.2 and earlier
 - AIX 4.3.3-5.2 (1)
 - AIX 4.3.3-5.2 (2)
 - AIX 4.3.3-5.2 (3)
 - AIX 5.3 ML1
 - AMIGA
 - AMIGA 3.9 BB2 with Miami stack
 - AMIGAOS
 - AMIGAOS 3.9 BB2 MiamiDX
 - AXIS
 - AXIS Printer Server 5600 v5.64
 - Alteon
 - Alteon ACEswitch
 - BSD/OS
 - BSD/OS 3.1-4.3 (or MacOS X 10.2)
 - BeOS
 - BeOS 5.0-5.1
 - BeOS 5.0.x
 - CacheFlow
 - CacheFlow CacheOS 1.1
 - CacheFlow CacheOS 4.1
 - Cisco
 - Cisco 12008
 - Cisco 7200, Catalyst 3500, etc
 - Cisco Content Engine
 - Cisco IOS 11.0

IpMorph Personality Catalogue

sinfp-latest.db

- CheckPoint
- Cisco
- Citrix
- Compaq
- D-Link
- DEC
- Digital
- Fortinet
- HP
- IBM
- KYOCERA
- Linksys
- Microsoft
- Windows
 - Windows
 - 9x
 - NT
 - 2000
 - 2003

xprobe2.conf

- Microsoft Windows 2000 Server Service Pack 1
- Microsoft Windows 2000 Server Service Pack 2
- Microsoft Windows 2000 Server Service Pack 3
- Microsoft Windows 2000 Server Service Pack 4
- Microsoft Windows 2000 Workstation
- Microsoft Windows 2000 Workstation SP1
- Microsoft Windows 2000 Workstation SP2
- Microsoft Windows 2000 Workstation SP3
- Microsoft Windows 2000 Workstation SP4
- Microsoft Windows 2003 Server Enterprise Edition
- Microsoft Windows 2003 Server Standard Edition
- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows 98 Second Edition (SE)
- Microsoft Windows Millennium Edition (ME)
- Microsoft Windows NT 4 Server
- Microsoft Windows NT 4 Server Service Pack 1
- Microsoft Windows NT 4 Server Service Pack 2
- Microsoft Windows NT 4 Server Service Pack 3
- Microsoft Windows NT 4 Server Service Pack 4
- Microsoft Windows NT 4 Server Service Pack 5
- Microsoft Windows NT 4 Server Service Pack 6a
- Microsoft Windows NT 4 Workstation
- Microsoft Windows NT 4 Workstation Service Pack 1

nmap-os-db

- 2Wire
 - 2Wire 1701HG wireless ADSL modem
 - 2Wire 2700HG, 2700HG-B, 2701HG-B, or RG2701HG wireless ADSL...
 - 2Wire 2701HG-B wireless ADSL modem
 - 2Wire Shasta wireless broadband router
- 3Com
 - 3Com OfficeConnect 3CRWE554G72T wireless broadband router
 - 3Com OfficeConnect 3CRWER100-75 wireless broadband router
 - 3Com OfficeConnect 3CRWER100-75 wireless router
 - Wireless broadband router (3Com OfficeConnect 3CRWDR100A-72, ...)
 - broadband router
 - switch
- ADIC
- storage-misc
- AKCP
 - AKCP sensorProbe2 remote sensor device
- APC
 - AOS
 - APC AP9319 Environmental Monitoring Unit or Smart-UPS 1000 RM ...
 - power-device
- AT&T
- Windows
- AVM
 - AVM Fritz!Box FON WLAN 7050, Linksys WAG200G, or Netgear DG8...

IpMorph : « vers la mystification de la prise d'empreinte de pile »

Personality Manager : Database Overview



XProbe2Signature
icmp_echo_reply_code: bool
icmp_echo_reply_ip_id: int
icmp_echo_reply_tos_bits: bool
icmp_echo_reply_df_bit: bool
icmp_echo_reply_ttl: int
icmp_timestamp_reply_received: bool
icmp_timestamp_reply_ttl: int
icmp_timestamp_reply_ip_id: int
icmp_address_mask_reply_received: bool
icmp_address_mask_reply_ttl: int
icmp_address_mask_reply_ip_id: int
icmp_information_reply_received: bool
icmp_information_reply_ttl: int
icmp_information_reply_ip_id: int
icmp_udp_unreach_reply_echoed_dtsize: int
icmp_udp_unreach_reply_ttl: int
icmp_udp_unreach_reply_precedence_bits: int
icmp_udp_unreach_reply_df_bit: bool
icmp_udp_unreach_reply_ip_id: int
icmp_udp_unreach_reply_payload_udp_chksm: int
icmp_udp_unreach_reply_payload_ip_chksm
icmp_udp_unreach_reply_payload_ip_df: bool
icmp_udp_unreach_reply_payload_total_len: int
icmp_udp_unreach_reply_payload_3bit_flags: bool
tcp_syn/ack_reply_tos: int
tcp_syn/ack_reply_df: bool
tcp_syn/ack_reply_ip_id: int
tcp_syn/ack_reply_ttl: int
tcp_syn/ack_reply_ack_delta: int
tcp_syn/ack_reply_wss: int
tcp_syn/ack_reply_options: byte[]
tcp_syn/ack_reply_wscale: int

Nmap2Signature
tcp_t1_ecn_reply: int[]
tcp_t2_reply: int[]
tcp_t3_reply: int[]
tcp_t4_reply: int[]
tcp_t5_reply: int[]
tcp_t6_reply: int[]
tcp_t7_reply: int[]
icmp_ie_reply: int[]
udp_ul_reply: int[]

SinFPSignature
ip_ttl: bool
ip_id: int
ip_df: bool
tcp_seq: int
tcp_ack: int
tcp_flags: byte[]
tcp_wss: int
tcp_mss: int
tcp_options: byte[]

p0fSignature
ip_dont_fragment: bool
tcp_wss: int
tcp_packet_size: int
tcp_initial_ttl: int
tcp_mss: int
tcp_window_scaling: int
tcp_options: byte[]
tcp_quirks: bool[]

Ring2Signature
nb_packets: int
ring_timeout: int
connexion_time: int
retry_intervals: int[]

myToolSignature
?
?
?



IpMorphPersonalityDatabase
[ipMorphPersonality.db]



IpMorph : « vers la mystification de la prise d'empreinte de pile »

IpMorph : Illustration pratique

DEMONSTRATION



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Etat d'avancement

- **Conception = 70%**
 - Etat de l'art bibliographique = 85%
 - Etat de l'art technique mystification = 70%
 - Analyse des outils d'OSFP = 70%
 - *Nmap* = 80%
 - *Xprobe2* = 95%
 - *SinFP* = 60%
 - *Satori* = 45%
 - *Ring2* = 80%
 - Architecture générale = 80%
 - Spécification & UML/Software design= 50%
- **Implémentation = 10%**
 - Proof of concept « trash programming » = 85%
 - Développement des composants = 5% (50% pour Personality Manager)
 - *Tests unitaires, tests fonctionnels, documentation* = 5%



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Perspectives

- **Juin 2009 – SSTIC 2009**
 - Beta release 0.5 en download
 - Site web, Mailing list, Bugtracking
 - Subversion / sources
- **Décembre 2009**
 - Version 1.0 en download
 - Documentation, UserGuide, DevGuide
 - Distribution .deb, .rpm, ...
 - Intégration de quelques scrubbers applicatifs (DNS, SMB, DHCP, ...).



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Bibliographie 1/2

1. **Smart, M., Malan, G.R., Jahanian, F.**: [Defeating TCP/IP Stack Fingerprinting](#), 9th USENIX Security Symp.
2. **Smith, C., Grundl, P.** : [Know Your Enemy: Passive Fingerprinting](#), <http://old.honeynet.org/papers/finger/>, (2002)
3. **Fyodor** : [Remote OS detection via TCP/IP Stack FingerPrinting](#), Phrack 1998, <http://www.insecure.org/nmap/nmap-fingerprinting-article.txt>, (1998)
4. **Spangler, R.**: [Analysis of Remote Active Operating System Fingerprinting Tools](#), <http://www.packetwatch.net/documents/papers/osdetection.pdf>
5. **Veysset, F., Courtay, O., Heen, O.**: [New Tool And Technique For Remote Operating System Fingerprinting](#), http://www.hackerz.ir/e-books/remote_os_detection.pdf, (2002)
6. **Auffret, P.** : [SinFP, Unification de la prise d'empreinte passive et active des systèmes d'exploitation](#), http://actes.sstic.org/SSTIC08/SinFP_Unification_Prise_Empreinte_Active_Passive_Systemes_Explotation/ , SSTIC 08, (2008)
7. **Berrueta, D.B.**: [A practical approach for defeating Nmap OS-Fingerprinting](#), <http://nmap.org/misc/defeat-nmap-osdetect.html> (2003)
8. **Crenshaw, A.** : [OSfuscate: Change your Windows OS TCP/IP Fingerprint to confuse P0f, NetworkMiner, Ettercap, Nmap and other OS detection tools](#), <http://www.irongeek.com/i.php?page=security/osfuscate-change-your-windows-os-tcp-ip-fingerprint-to-confuse-p0f-networkminer-ettercap-nmap-and-other-os-detection-tools>, (2008)
9. **Provost, N.** : [Honeyd: A Virtual Honeypot Daemon](#), <http://www.citi.umich.edu/u/provos/papers/honeyd-eabstract.pdf> , (2003)
10. **Wang, K.** : [Frustrating OS Fingerprinting with Morph](#), <http://www.synacklabs.net/projects/morph/Wang-Morph-TheFifthHOPE.pdf>, The Fifth HOPE, (2004)
11. **Veysset, F., Courtay, O., Heen, O.**: [Détection des systems d'exploitation avec RINGv2](#), Actes SSTIC, (2003)
12. **Fusys,CyRaX** : [Fingerprint Fucker](#), <http://www.s0ftpj.org/tools/fingfuck.tgz>
13. **Reed, D.** : [Fingerprint Fucker](#), <http://packetstormsecurity.org/UNIX/misc/bsdfpf.tar.gz>
14. **Trifero, S., Callaway, D.** : [Linux Kernel Stealth Patch](#), <http://www.innu.org/~sean/>



IpMorph : « vers la mystification de la prise d'empreinte de pile »

Bibliographie 2/2

15. **Aubert, S.** : [Antimap](http://www.hsc.fr/ressources/breves/antimap.html), <http://www.hsc.fr/ressources/breves/antimap.html>
16. **FreeBSD, Blackhole**, <http://www.gsp.com/cgi-bin/man.cgi?section=4&topic=blackhole>
17. **McCabe, R.** : [Iplog](http://ojnk.sourceforge.net/stuff/iplog.readme), <http://ojnk.sourceforge.net/stuff/iplog.readme>
18. **OpenBSD : PF « le filtrage de paquet »**, <http://www.openbsd.org/faq/pf/fr/filter.html>
19. **Roualland, G., Saffroy, J-M.** : [IP Personality](http://ippersonality.sourceforge.net/), <http://ippersonality.sourceforge.net/>
20. **Wang, K.** : [Frustrating OS Fingerprinting with Morph](http://www.synacklabs.net/projects/morph/Wang-Morph-DEFCON12.pdf), DEFCON 12,
<http://www.synacklabs.net/projects/morph/Wang-Morph-DEFCON12.pdf> (2004)



Merci de votre attention.

Interrogations ? Critiques !

Suggestions...



Nous contacter :
contact@ipmorph.org
Beta Release 2009-06

<http://dev.ipmorph.org>