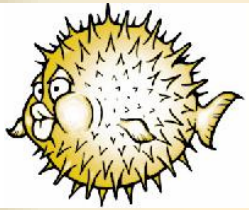
OpenBSD

Can E. Acar, Berk D. Demir

canacar@openbsd.org

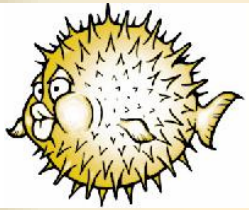
bdd@mindcast.org

OpenBSD != Linux



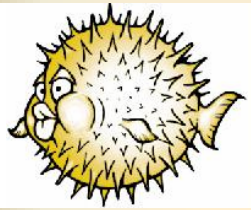
OpenBSD

- Özgür, Açık, Bedava
- Çok platformlu
- 4.4BSD tabanlı
- Taşınabilir
- Standart
- Doğru



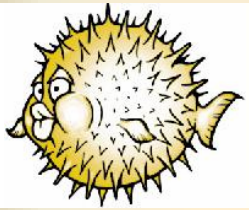
Farklı Yapılanma

- Bir hobi projesi
  - Hobi != Ciddiyetsizlik, Hobi != Kalitesizlik
  - Marketing yok. Müşteri kaygısı yok
  - Hedefler var
- Kapalı bir grup
- OpenBSD geliştirici için “Free” nin anlamı
  - NDA imzalanmaz
  - Özgürlüğün “ama” sı olmaz. (TRUE or FALSE)
- Elitist, Akademik
  - Biraz da meritocracy



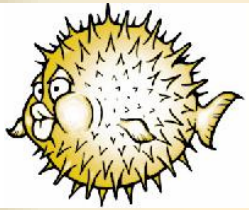
Farkı Yapılanma - II

- Sadece çekirdek değil, tam bir işletim sistemi
  - Çekirdek
  - ‘Base system’
    - Temel UNIX komutları
    - Apache, Bind, X, SSL, ssh, perl, gcc, kerberos ...
    - … ve mümkünse GPL lisanslı yazılım bile içermesin
  - Dokümantasyon (man pages)
- Ports
  - Üçüncü parti yazılımlar.
  - En doğru şekilde yapılandırılmış.
  - Sisteme uyum sorunları giderilmiş, önemli güvenlik açıkları kapatılmış.
  - Müdahale gerektirmeyen derleme ve kurulum süreci
  - Binary paketler halinde dağıtım ve kurulum imkanı
  - Yaklaşık 2500 port



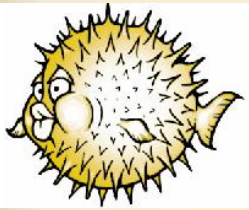
Farklı Yapılanma - III

- Düzenli çıkan sürümler
  - Her altı ayda bir yeni bir sürüm
- Yeni özelliklerden çok, mevcut özelliklerin geliştirilmesi
- Dokümante edilmemiş veya eksik dokümante edilmiş yazılım dağıtıma katılmaz



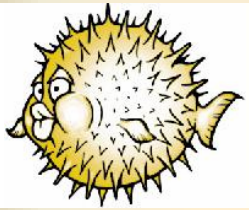
Tarihçe

- 1987 : 4.4BSD
  - Kaliforniya Üniversitesi, Berkeley
  - Bilimsel, UNIX benzeri, açık kaynak kodlu, askeri destekli.
- 1993 – 94 : NetBSD ve FreeBSD projelerinin doğuşu
  - FreeBSD
    - Intel x86 mimarisi
    - Yüksek performanslı sunucu
  - NetBSD
    - Çok platformlu
    - Temiz, taşınabilir kod
- 1996: NetBSD projesi içinden OpenBSD



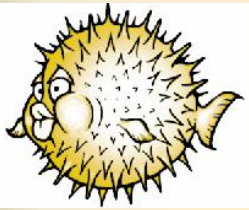
Hedefler

- Açıklık
  - Açık kaynak, açık CVS ağacı
  - Herkesin, her amaçla kullanabilmesi
- İyi parçaları entegre etmek
  - Kabul edilebilir lisanslı her parçayı
- Güvenlik
  - Açıkları çıkmadan önce yok etme
  - En güvenli genel amaçlı işletim sistemi olma
    - Eğer hala değilsek?
  - **Varsayılan kurulumda, 7 yıldır, sadece bir uzaktan güvenlik açığı**



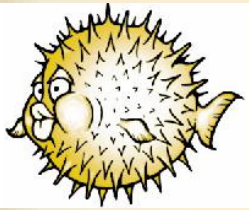
Hedefler

- Entegre Kriptografi
  - IPSec, IPv6, Key Servers, Kerberos, OpenSSH, vb.
- Standartlar
  - POSIX, ANSI, X/Open
- Olabildiğince platform bağımsız kaynak kodu
- Politikadan uzak, teknik değer üreten çözümler
- Ciddi problemler çözümsüz kalmaz



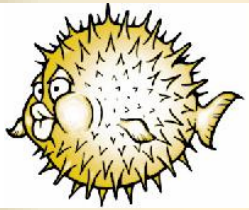
Proaktif Güvenlik

- Tüm kaynak ağacının taranması
  - Güvenlik açığı değil, yazılım hatası aramak
- Bir hata türü bulunduğunda, tüm kaynak ağacı üzerinde tarama
  - format string, realloc, string fonksiyonları, vb.
- Olası güvenlik açıklarını önceden tespit etmek ve engelleyici önlemler almak
  - Propolice, W^X, Privilege Seperation, vb.



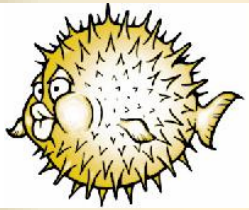
Entegre Kriptografi

- OpenSSH, IPSec, Kerberos
- İlk entegre kriptografi kullanan Unix benzeri işletim sistemi
  - Amerikan ihracat yasalarından etkilenmeyen, Kanada tabanlı proje
- Donanımsal kriptografi desteği
  - HIFN ve benzer kriptografi hızlandırıcıları
  - Kullanıcı programlarının desteklenen donanımları bir değişiklik gerekmeden kullanabilmesi



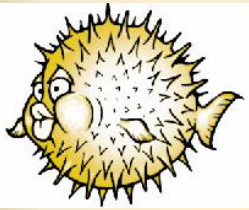
Saf Unix

- POSIX uyumluluğu önemli bir esastır.
- ANSI ve X/Open standartlarına uyulur
- Her yeni özellik, her yeni düzeltme Unix uyumlu ise geçerlidir.
- OpenBSD uyumlu olması YETERLİ DEĞİLDİR!



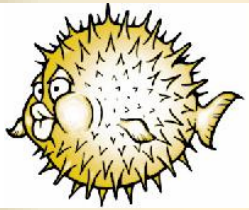
Üretilen Değerler

- OpenSSH
- PF ve ALTQ
- Olgun IPSec ve IPv6 yığıtı
  - KAME tabanlı
  - Cisco, Lucent, Nortel, VPNC
- Buffer overflow saldırıları ile mücadele
  - Non-exec stack, Non-exec heap
  - Propolice destekli GCC
  - Write XOR Execute (W^X), !W, !X
  - malloc / mmap randomization
  - Guard Pages, vb.



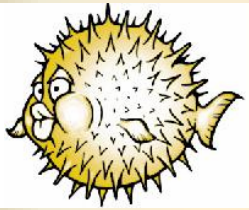
Üretilen Değerler

- AnonCVS
  - CVS ağacına salt okunur, anonim erişim
- authpf
  - Kullanıcıya göre belirlenen güvenlik duvarı kuralları
  - VPN'ler
- spamd
  - Spammerlara pasif karşı saldırı
- privdrop
  - Programların yetkilerini en kısa sürede bırakması
  - neredeyse bütun s[ug]id programlar, ftpd, httpd, named, ...
- privsep
  - Yetki gereken işlemlerin küçük ve izole program parçasında gerçekleştirilmesi
  - sshd, syslogd, pflogd, isakmpd, XServer, xterm, ...



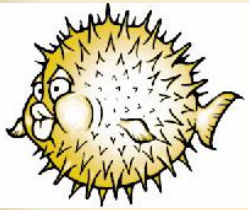
OpenSSH

- Güvenli uzaktan terminal bağlantıları
  - Port yönlendirmesi ile güvenli tüneller
  - X Window System için yönlendirme
  - Entegre SOCKS Proxy
- İlk SSH gerçekleştiriminin lisansının değişmesi
  - Sadece ticari olmayan kullanım için bedava
- En son özgür sürümden ortaya çıkan OpenSSH
- SSH1 ve SSH2 protokol destekleri
  - Tek programda
- Privilege Seperation



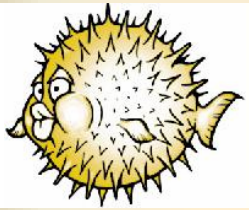
Herkes İçin OpenSSH

- IBM, AIX
- Sun Microsystems, Solaris
- Cisco
- Alcatel
- Hewlett Packard
- Lucent
- Siemens
  - Telefon Santrali



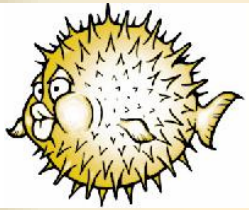
OpenBSD PF

- Eski paket filtresi IPF'in lisansının değişmesi
- Daniel Hartmeier bu kaos içinde PF'i gerçekleştirir
- IPF ile söz dizimi uyumlu (gibi...)
- Gerçek anlamda *Stateful inspection*
  - Guido van Rooj
  - http://madison-gurkha.com
- 30'dan fazla geliştirici



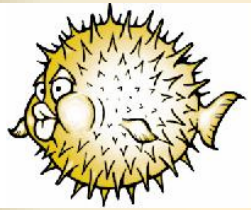
OpenBSD PF

- Scrubbing
- NAT/redirection
- Bridge modunda filtreleme
- Ayarlanabilir zaman aşımı süreleri
- IPv6
- Anti Spoof
- ALTQ entegrasyonu
- Adres tabloları
- Anchor (alt kurallar)
- Yük dengeleme (load balancing)
- Paket etiketleme (tagging)
- Pasif işletim sistemi tespiti (p0f v2)



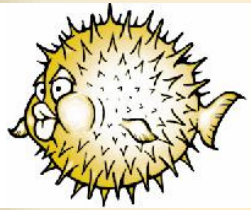
ALTQ

- OpenBSD 3.0’dan bu yana
- Trafik yönetimi
  - CBQ
  - PRIQ
  - HFSC
  - RED, ECN, RIO
- OpenBSD 3.3 ile gelen PF entegrasyonu
- *Stateful inspection* sayesinde gelen paketlerin çıkışını şekillendirme



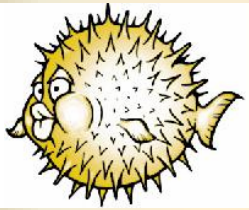
PF - Adres Tabloları

- Çok büyük adres listeleri için çok hızlı arama
- Ağ bloğu veya ağ adresi verebilme
  - ...ve bunları tek tek tersleyebilme
- Dinamik olarak içeriği değiştirilebilir
- OpenBSD 3.3 ile
- Radix Tree, O(1)
- Kullanım alanları
  - spamd
  - Saldırı tespit ve tepki



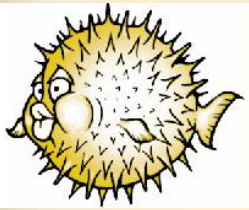
PF - Anchor (Alt Kurallar)

- Ana kural seti içinden çağırılan kural kümeleri
- Dinamik olarak içeriği değiştirilebilir
- Kullanım alanları
  - authpf
  - Saldırı tespit ve tepki
- OpenBSD 3.2 ile



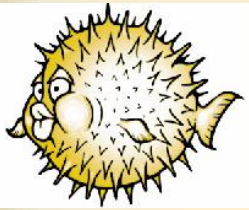
IPSec ve IPv6

- IPv6
  - KAME tabanlı
  - 1999 yılında, 2.6 sürümünden beri
  - En olgun IPv6 yığıtlarından biri
  - Kullanıcı uygulamalarının büyük bir bölümü IPv6 destekli.
- IPSec
  - VPN'lerin vazgeçilmez protokolü
  - IPSec yığıtı entegre ilk Unix benzeri işletim sistemi
  - 2.6'dan beri
  - %100 VPNC uyumlu



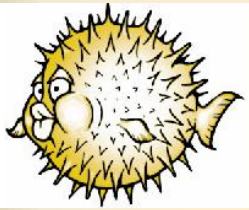
Bilgiye Ulaşmak

- Man Sayfaları
  - OpenBSD != Linux, Solaris, HP-UX, ...
  - Geliştiriciler kod yazmaya ayırdıkları kadar zamanı man sayfalarına da ayırır
- OpenBSD SSS
  - http://www.openbsd.org/faq
  - Laf salatası yok
- E-Posta Listeleri
  - http://www.openbsd.org/mail.html
  - Arşivler: MARC, Google, vb.



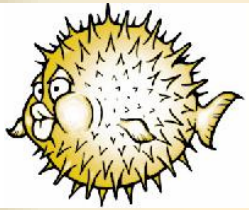
OpenBSD'ye Ulaşım

- Web sayfası
  - **http://www.openbsd.org**
  - **http://www.tr.openbsd.org**
- FTP
  - **ftp://ftp.openbsd.org/pub/OpenBSD**
  - **ftp://ftp.linux.org.tr/pub/OpenBSD**
- RSYNC
  - **http://www.openbsd.org/ftp.html#rsync**
  - **rsync://ftp.linux.org.tr::OpenBSD**
- Diğer
  - HTTP, AFS
    - **http://www.openbsd.org/ftp.html**
  - AnonCVS, CVSup, CTM



İlerideki Hedefler

- CARP
  - Patent problemi olan HSRP ve VRRP router redundancy/failover protokollerine alternatif.
  - `3.4-current` içerisinde hazır
- PF state senkronizasyonu
- Daha çok privsep (dhcp)
- PowerPC mimarisi için W^X
- Yeni mimariler Pegasos-PPC, ARM, ...
- Geliştiricilerin canı ne isterse...



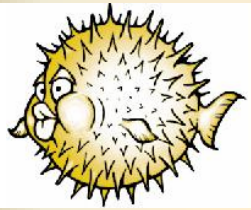
Nasıl Yardımcı Olabilirsiniz?

- Kod yazarak
- Hata bularak
- Dokümantasyon
- Maddi katkı
  - CD satışları
  - T-Shirt
  - Poster
  - Bağışlar
  - Donanım bağışları

10 Dakika Ara

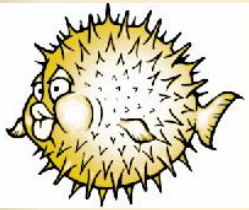
Hemen Ardından

OpenBSD'de
Suistimali Zorlaştıran
veya
İmkansız Kılan Teknikler



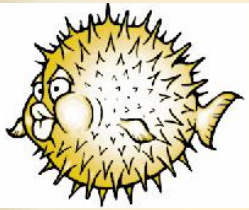
Hedefler

- Unix programları POSIX ortamında çalışır ve POSIX üç şeyi belirtir
  - Unix'de değiştirememeniz gereken şeyler
  - Unix'de değiştirebileceğiniz şeyler
  - Unix'de … tanımsız bırakılmış şeyler
- Programların beklediği davranışı bozma
- Programların bel bağladığı davranışı KESİNLİKLE bozma
- Exploit yazarını ağlatabilecek her şeyi uygula.



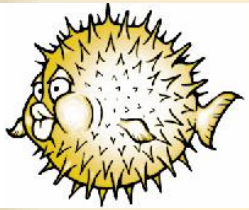
Stack Smashing

- Klasik tampon bellek (stack) taşması suistimali:
  - Stack üzerinde bulunan bir tampon belleğin içerisine (programlama hataları nedeniyle) alabileceğinden daha fazla veri yazılması.
  - Fazla gelen veriler bellekte daha sonra gelen değişkenlerin ve en önemlisi fonksyonun dönüş adresinin üzerine yazılır.
  - Saldırgan fazla verinin yazılmasına yol açacak hatayı kullanarak ve yazılan verileri kontrol ederek, kendi belirlediği (belleğe yerleştirdiği) bir kodun çalışması sağlanır.
- Örnek: sayılamayacak kadar çok



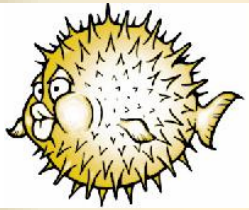
Propolice / SSP

- Stack-smashing saldırılarını yakalayan GCC eklentileri
  - IBM'in projesi
    - http://www.trl.ibm.com/projects/security/ssp
  - OpenBSD kaynak ağacına eklendikten sonra 26+ adet hatası düzeltildi
- Propolice'li GCC ile normal GCC aynı kalitede *(eğer kaliteden söz edebiliyorsak)*
- StackGuard'dan çok daha iyi
  - Core Security Tech. incelemesi
  - En azından sadece i386 için değil



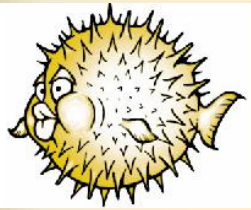
Propolice / SSP

- Nasıl çalışıyor?
  - Stack'e bir kanarya yerleştiriliyor
  - Bu kanarya fonksyondan geri dönüşte kontrol ediliyor
  - Kanarya değiştiyse problem var
  - İşletim durduruluyor
- Kanaryayı doğru yere yerleştiriyor
- Stack'deki nesneleri güvenlik için yeniden sıralıyor
- Zayıflığı: Fonksiyonlar için korumanın atlanması algoritması çok iyi değil. Üzerinde çalışma devam ediyor



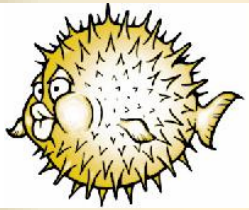
Propolice / SSP - Sonuç

- Güvenliğe katkısı
  - Hataları bulur
  - Bu hataların suistimal edilmesini imkansız kılar
- Çok düşük maliyet
  - Herkes kullanmalı
  - Kullanmamak için hiçbir mantıklı sebep yok
- Performans
  - Çok az performans kaybı: yaklaşık %1-2



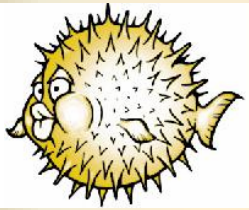
Kod Çalıştırılmasının Engellenmesi

- Saldırgan *bir şekilde* programın akışını değiştirebilir.
  - Stack/Heap üzerindeki program adresleri ve kontrol yapıları
  - Format string zayıflıkları
  - ...
- Programın akışı değiştiğinde saldırganın istediği kodu çalıştırması gerekir. Çalışacak bu kodun programın bellek bölgesi içerisine yerleştirilmiş olması gerekir:
  - **Program alanı**: yazılamaz, ancak mevcut kodlar bazen saldırganın işine yarayabilir
  - **Stack alanı**: yazılabilir ve kolayca veri/kod yerleştirilebilir.
  - **Data alanı**: yazılabilir ve genellikle veri/kod yerleştirmesi kolaydır (heap).
- Stack ve Data alanı için çözüm: W^X



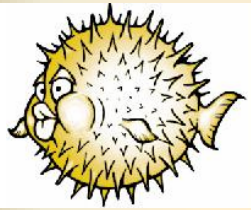
W^X Mekanizması

- Bellek sayfalarının hem yazılabilir hem de işletilebilir olması bir çok hatanın süistimal edilebilmesinin temel sebebi
- Bir bellek sayfası ya sadece yazılabilir olmalı ya da sadece işletilebilir.
- Buna Write XOR Execute diyoruz
  - “Ya Yaz Ya da İşlet”



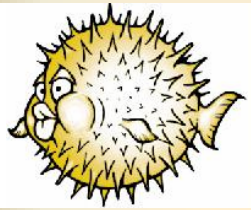
W^X ve Mimariler

- Her mimaride mümkün değil
- Tek bir yöntem ile tüm mimarilerde çözüm getirmek de mümkün değil
  - **`sparc, sparc64, alpha :`** Per Page X-bit
  - **`hppa :`** Per Page X-bit
  - **`i386 :`** Code Segment Limit
  - **`m88k :`** Per Page X-bit (*devam ediyor)*
  - **`powerpc :`** Per Segment X-bit *(devam ediyor)*
  - **`mips :`** Henüz mümkün değil *(**`tlb`** ile oynasak?)*
  - **`vax, m68k :`** Mümkün değil



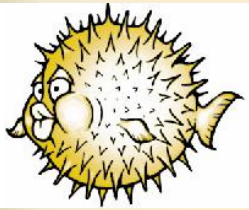
W^X Gerçekleştirimi
sigtramp'in yerinin değiştirilmesi

- Stack, kötü niyetli kodların yerleştirebileceği yerlerden birisi.
- **Problem**: Stack'in ilk sayfasında çalıştırılabilir olması gereken *Signal Trampoline* bölgesi var.
- **Çözüm**: sigtramp'i, stack'in dışarısına çıkartıp, **`R-X`** hakkı veriyoruz. Stack **`RW-`** hakları ile kalıyor.
- Puff! Stack üzerinde kod işletmek artık mümkün değil



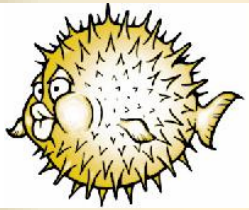
W^X Gerçekleştirimi GOT/PLT/dtors

- Kısaltmalar
  - GOT: Global Offset Table
  - PLT : Procedure Linkage Table
  - ctors/dtors: Constructors / Destructors
- Data segment'i içinde yer alan PLT işletilebilir olmalı; ama data segment'i işletilmemeli
- Data segmenti yazılabilir olmalı; ama GOT, PLT ve .dtors yazılabilir olmamalı



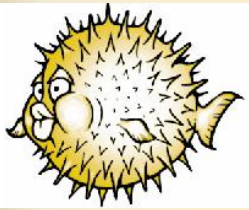
W^X Gerçekleştirimi GOT/PLT/dtors

- Per Page X-bit destekleyen mimarilerde
- GOT ve PLT kendi sayfalarına sahip oluyor. Bu sayede artık yazılamaz oluyorlar.
  - `ld.so` da bu yeni sistemi anlayacak şekilde eğitildi
- ctors ve dtors da GOT sayfası içinde
- Eh hayırlı olsun!
  - Data segment'i işletilebilir bir nesne taşımıyor artık



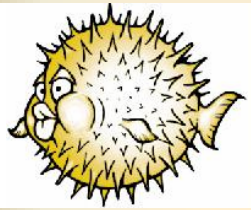
W^X Gerçekleştirimi GOT/PLT/dtors

- Per Page X-bit desteklenmiyorsa?
- O zaman “range of execution”
  - i386 (RoE) ve PowerPC (per segment x-bit)
- Veriyi ve Kodu ayır
- Arasına bir çizgi çek
- Üst taraf (data) işletilebilir değil
- Alt taraf (kod) yazılabilir değil



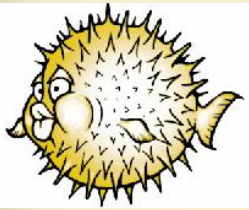
W^X Özet

- sigtramp stack'in dışına çıkıyor
- GOT/PLT kendi sayfalarına sahip
- ctors/dtors başka sayfalarda
- Ama i386 PP X-bit desteklemediği için
  - Code segment limiti 512MB'da
  - Tüm işletilebilir parçalar aşağıda, yazılabilirler yukarıda
- PowerPC i386'ya benzer durumda ama biraz daha kompleks



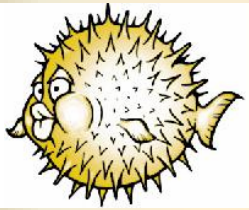
W^X Sonuç

- Performans
  - Bazı mimarilerde hızlanma (TLB renklenmesi)
- Güvenlik
  - W^X desteği bir çok güvenlik açığını tek başına ortaya çıkarttı
- Düşük Maliyet
  - Tüm işletim sistemi üreticileri buna yönelmeli



`.rodata` Segmenti

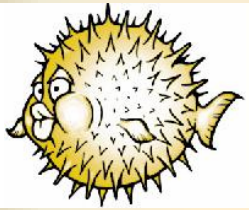
- `!X` de diyebiliriz
- Salt okunur veriler ve işaretçiler .text alanında saklanıyordu. Okunabilir ve işletilebilirlerdi.
- Yani `const` data işletilebilir. Saldırganın kullanabileceği bir kod olabilir.
- ELF için `.rodata` diye bir segment yaratıldı ve sadece `PROT_READ` verildi.
- Artık bu veriler `.rodata`'da
- Hiçbir maliyeti yok.
- Minimum yetki ile çalışmanın güvenilirliği



Rassal Bellek Tahsisi

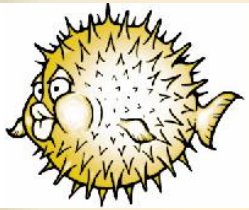
- Randomized malloc()

```
p = malloc(16);
free(p);
p2 = malloc(16);
if (p != p2)
    Mükemmel!
```



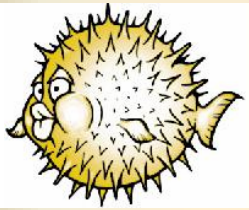
Koruma Sayfaları

- Elektrikli tel gibi.
  - Dokunursan çok canın yanar
  - Becerebiliyorsan dokunmadan üzerinden atla.
- Her bellek tahsisinin arasını bir tane haritalanmamış sayfa koy
- Bu sayfaya denk gelen program bellek hatası verip çıkacaktır.
  - Normal çalışan bir program kendi belleğini taramaz
  - Hatalı yazılmış kod veya bir exploit'den başkası bu sayfalara rastlamaz
- Sanal bellekte çok az bir alan harcıyor



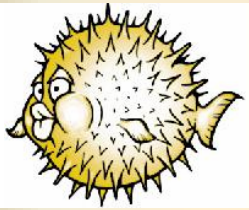
Stack Gap Randomization

- Stack'in rassal 8 byte hizalanması
- En fazla 1 sayfa gerçek bellek harcıyor
- random(N) kadar süreç belleği harcıyor
- Sistem yöneticisi mesafeyi ayarlayabilir



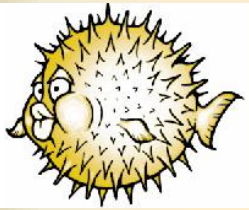
Shared Lib. Sırası ve Rassal Erişim

- Paylaşımlı kütüphanelerin dinamik yüklenme esnasında yüklenme sırasını rastlantısal olarak belirleniyor
- Önceden bilinen paylaşımlı kütüphaneler için rastlantısal olarak haritalama adreslerinin belirlenmesi
- Sonuç:
  - Çok ucuz, saldırgan için acı verici
  - Hatasız değil ama atlatması çok zor
  - Bu değişiklikler hiçbir yazılımı etkilemez



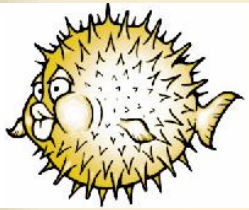
Yetki Bırakma (privdrop)

- set[ug]id programlar veya sunucu için
- Yetki bırakma
  - Yetki gerektiren işlemleri gerçekleştir: **`/dev/pf`**, **`/dev/bpf`**, özel soketler, utmp vb.
  - Mümkünse **`chroot()`** ile çalışma alanını kısıtla
  - Yetkileri bırak
- Basit programlar için
  - ping, ping6, portmap, rpc.statd, rpc.rusersd, traceroute, traceroute6, rwalld, pppd, spamd, authpf, ftpd, named, httpd
- Geliştilmesi oldukça kolay



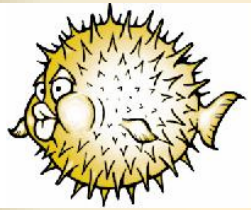
Yetki Ayrıştırma (privsep)

- Yetki Ayrıştırma
  - Yetki gerektiren ve baştan gerçekleştirilebilecek işlemler gerçekleştirilir.
  - Haberleşme için bir **`socketpair()`** yaratılır ve **`fork()`** ile işlev ikiye bölünür.
  - *Büyük* işlev **`chroot()`** ile kendisini sınırlandırıp yetkilerini bırakır.
  - *Küçük* işlev yetkilerini elinde tutar.
  - Program *büyük* süreçte çalışmaya devam eder. Yetki gereken işlemleri *küçük* süreçten ister
- Karışık ve zor bir tasarım ve programlama gerektirir
  - sshd, syslogd, pflogd, isakmpd, Xserver, xterm, xdm, xconsole
- httpd ve dhcp programları için privsep çalışmaları devam ediyor.



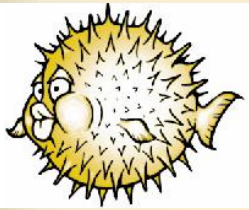
Özet

- Saldırganın karşılaştığı zorluklar
  - Frame pointer'in veya geri dönüş adresinin üzerine yazmak için 32 bitlik bir sayıyı tahmin etmesi gerekiyor
  - Bayraklar ve işaretçiler (flag/pointer) stack'in altında
  - Yazılabilir adres alanından hiçbir şey işletilebilir değil
  - **`signal()`** trambolini yazılabilir değil
  - GOT, PLT ve dtor yazılabilir değil
  - **`const`** data işletilebilir değil



Özet (devam)

- Saldırganın karşılaştığı zorluklar
  - Paylaşımlı kütüphaneler (shared libs) her seferinde farklı bir adrese haritalanıyor
  - malloc() ve mmap() rastlantısal tahsis yapıyor
  - malloc() ve mmap() guard page yerleştiriyor (çalışma devam ediyor)
  - Stack'in tepesi rastlantısal olarak dengeleniyor (ve hizalanıyor)
  - Bir çok yetki kullanan program ve sunucu bu yetkilerini bırakıyor (privdrop)
  - Diğer sunucular ve programlar yetkilerini başka süreçlere aktarıyor (privsep).



Sonuç

- Ucuza mal olan teknikler
  - Önemsenmeyecek performans azalması
- Neredeyse Evrensel
  - Önemli bir bölümü tüm mimarilerde çalışıyor.
  - İyi mimariler için toplam kapsama
- POSIX semantiklerine çiğnememe
  - Tüm yazılımlar eskisi gibi çalışmaya devam ediyor
- İşletim sisteminizi yazanlardan bu özellikleri isteyin.
  - Çünkü bu sizin hakkınız

Teşekkürler

Can E. Acar <canacar@openbsd.org>

Berk D. Demir <bdd@mindcast.org>