逆引き shellshock

やまぐちたかのり

shellshock とは?

- bash には、POSIX sh からの拡張機能として、シェル関数のエクスポート機能があった
  - 環境変数経由でエクスポートされる

```
export -f function_name
```

- parser がいいかげんで、細工された環境変数を取り込むとコード実行してしまう

```
HOGE="() { : function body;}; : injection code"
```

  - 修正したと思ったら別のパターンでコード実行可能になったりして、短期間で CVE 連発
- 外部から任意の環境変数をセットできる状況でとくにヤバい
  - 典型例: HTTP リクエストヘッダを環境変数にセットして CGI を呼ぶ

DNS もヤバい?

- REMOTE_HOST って環境変数に心あたりあるよね?
- IP アドレスを逆引きしたホスト名を環境変数にセットしている場合、そこからコード注入のおそれが
  - http://seclists.org/fulldisclosure/2014/Oct/53
  - https://www.nic.ad.jp/ja/topics/2014/20141017-01.html
- でも、“`() { :;}; : injection code`” なんてホスト名ってDNS 的にアリなの?

DNS の仕様

- ホスト名として使える文字は、英数ハイフンのみ(RFC1123)
- が、ホスト名のデータベースとしての DNS には、そのような制限はない
  - RFC2181 section 11 に、「ラベルには長さ以外の制限はない」と明記されている
  - 長ささえ制限以下なら、バイナリだろうがなんだろうが OK
    - アプリケーションがその名前を正しく扱えるかどうかはまた別の話
- shellshock の攻撃コードを DNS に登録するのも、仕様上まったく問題ない
- DNS の仕様は守ってくれない

DNS サーバの実装

- 権威サーバ
  - BIND は、おかしな名前をゾーンに書くとエラーになる
    - が、`options { check-names master ignore; };` でおかしな名前も許容させることができる
  - NSD はとくにチェックしない
  - こんなふうに書けばおっけー

```
1.0.254.169.in-addr.arpa. IN PTR \(\)\ {\ :\;}\;echo\ hoge.
```

- キャッシュサーバ
  - BIND、Unbound とも、権威サーバが返してきた結果をそのままクライアントに返す
  - ホスト名としては RFC1123 的に不正でも、DNS 的には valid な名前なので、よけいなおせっかいはしない
- DNS サーバは守ってくれない

リゾルバの実装

- dn_expand(3) の中でヤバい文字をエスケープ処理してる
  - DNS メッセージ中の圧縮された名前を伸長する関数
    - 実際にはさらにその下請け関数 ns_name_ntop() で処理
    - BIND8 由来
  - 一部の記号(`".;\()@$`)と 0x00-0x20, 0x7f-0xff の文字
    - ホスト名に使えない文字であっても、大半の記号は対象外
    - ゾーンファイルの記述に使われる記号だけが対象みたい
  - /etc/hosts など、DNS 以外で名前解決される場合は対象外
- libc (libresolv) は守ってくれた
  - libc 以外の実装は……?
  - 万が一エスケープ対象外の文字だけで可能な攻撃手法が発見されたら libc でもヤバい

OSX ヤバい? (1)

- full-disclosure に流れたメール
  - “At this point of time the stock resolvers (in combination with the libc library) of OSX 10.9 (all versions) and 10.10/R2 are the only known standard installations that pass the bash exploit string back and up to getnameinfo(). “
    - http://seclists.org/fulldisclosure/2014/Oct/53
- 試してみるとたしかにサニタイズされない
- が、オープンソースになってるコードを調べてみると、他の UNIX 系 OS と同様に、dn_expand() で記号その他を潰してる
  - https://opensource.apple.com/source/libresolv/
- どういうこと?

OSX ヤバい? (2)

- どうやら、OSX の getnameinfo(3) や gethostbyaddr(3) といった関数は、そもそも res_query() や dn_expand() を使ってないっぽい
  - getnameinfo() などの関数はオープンソースになっておらず、他の OS とは内部的な実装が大きく異なると思われる
  - これらがサニタイズをしていない模様
    - ただ、しないからといって脆弱性とかバグとか呼ぶのは微妙な気が
- res_hogehoge() や dn_hogehoge() で直接 DNS に問い合わせれば、アップデート適用前の環境でも脆弱性の影響を受けないのを確認
- OSX 独自の API (NSHost、CFHost) の挙動は未確認

OSX 以外は大丈夫なの?

- FreeBSD、NetBSD、Linux (glibc) は問題ないのを確認
- Windows がまったくサニタイズしない模様
  - bash の穴を突かれることはないだろうけど…
- そして…

OpenBSD

- dn_expand() でサニタイズ処理してない
  - res_hogehoge()、dn_hogehoge() で DNS に直接問い合わせるプログラムはホスト名がサニタイズされない
  - OpenBSD のリゾルバは近年大きく書き換えられたが、この部分だけは BIND4 由来のコードが取り残されている
- getnameinfo() や gethostbyaddr() などではかなり厳密にチェックされる
  - 記号をエスケープするのではなく、名前解決に失敗する
- まあ、デフォでは OpenBSD に bash は入ってませんが

攻撃ベクトル

- 外部からの入力をそのまま環境変数につっこんでいる
  - この場合は DNS の逆引き
- 子プロセスとして脆弱性のある bash を起動する
  - /bin/sh が bash の場合は、system(3) や popen(3) などの呼び出しでも間接的に bash が呼ばれる
- 以上の条件を満たすとコード注入が成功する
  - いろいろ検証してみた

CGI

- 接続元ホスト名を REMOTE_HOST にセットする
- apache (OSX) で注入に成功
  - OSX で外部からアクセスできる CGI を動かしてる例は多くないかと
  - apache 以外の実装は未確認
- リクエストヘッダから注入した方がずっと楽なので、わざわざ逆引きをいじらなくても……
  - WAF や IDS に気付かれにくいというぐらいか

tcsh

- 接続元を REMOTEHOST にセットする
  - csh にはない、tcsh の拡張機能
- 試してみたけど注入できなかった
  - ソース見るとできそうなんだけどなぁ…
  - うまく細工すれば注入できるかも?
- …と思ったら、環境変数 HOST (自分のホスト名)から注入できた
  - あやしげな野良 WIFI に OSX を接続している状態で tcsh を起動すると HOST に危険な文字列が混入
  - シェルスクリプトを実行すると乗っとられる

tcpserver

- djb によるスーパーデーモン(inetd/xinetd の親戚)
- 接続元を TCPREMOTEHOST にセットする
- 名前解決は libc を使わず自前実装しているが、適切にエスケープ処理しているので影響なし

- 別の作者による類似品である tcpsvd も、名前解決部分を tcpserver からパクってるので影響なし

tcp-env

- inetd から呼ばれたプログラムに tcpserver 同様の環境変数をセットするためのラッパー
  - qmail に含まれている
- res_query() などを直接叩いているので、OpenBSD でコード注入に成功
  - OpenBSD で inetd + tcp-env から #!/usr/local/bin/bash なスクリプトを呼ぶ状況を想像できない…

couriertcpd

- これもスーパーデーモン
  - courier-imap、courier-mta に含まれている
- 接続元を TCPREMOTEHOST にセットする
- 名前解決は libc を使わず自前実装で、かつ、記号類のエスケープをしていないので、OSX や OpenBSD 以外でもコード注入できる
  - debian で実際にコード注入できることを確認
  - paranoid チェックを通す必要あり

```
1.2.0.192.in-addr.arpa. IN PTR (攻撃コード).example.com.
(攻撃コード).example.com. IN A 192.0.2.1
```

- couriertcpd から bash を呼ぶような使い方は一般的ではない
- -nodnslookup という引数で回避可能

IDENT もヤバい

- サーバがクライアントに対して、接続してるユーザ名などの情報を問い合わせるプロトコル
  - 113/tcp (RFC931/1413)
  - 詳細はぐぐってください
- 環境変数 REMOTE_IDENT や TCPREMOTEINFO など
- ぶっちゃけ、現状では死んだプロトコル
  - が、前世紀にはそれなりに使われていたので、サポートしているものも多い
- HTTPD のアクセスログの第2カラムは、実は ident で拾った情報を記録するためのフィールドだったりする

IDENT の攻撃ベクトル

- sendmail、proftpd
  - IDENT 検索の機能はあるが、環境変数にセットしないので影響なし
- tcp_wrapper (/etc/hosts.allow)、apache (mod_ident)、tcpserver、tcp-env、tcpsvd
  - 設定により IDENT で取得した値を環境変数にセットすることは可能だが、サニタイズしていて注入不可
- couriertcpd
  - 環境変数TCPREMOTEINFO に IDENT で取得した値をセットするが、サニタイズしておらず攻撃コードの注入に成功
  - -noidentlookup で回避可能

shellshock だけじゃない

- たとえばクライアントの逆引きホスト名を HTML に埋めこんでる Web サイトがあったとして、こんな逆引きを登録してアクセスすると…

```
1.2.0.192.in-addr.arpa. IN PTR <script>alert\(\“hoge\”\)</script>.
```

- SQL インジェクションなどにも注意
- 攻撃者が直接コードを送り込むわけではないので、WAF や IDS では検知しづらい

- libc のリゾルバでサニタイズされる記号は一部のみ
- Windows はまったくサニタイズされない
- リゾルバに頼らずアプリケーション側でチェックを

まとめ

- たいていの OS なら、libc の名前解決関数が危険な文字を排除するので逆引き shellshock の影響は受けない
  - OSX や OpenBSD は使用する関数によっては影響を受ける
  - 名前解決を libc に頼らず自前実装しているものは要注意
- DNS 逆引きだけでなく IDENT も攻撃ベクトルになりうる
  - っていうか、とっとと bash をアップデートしろ

  - が、逆引きホスト名からの攻撃は bash 以外にもありうる
  - 「外から来た値を信用してはいけない」の鉄則にしたがうべし