Internet Week 2024 DNSOPS.jp BoF

自動化で変わったBINDとの向き合い方

～例の有償ソフトでDNS構築してみた その後～

KDDI株式会社
エンジニアリング推進本部　プラットフォームエンジニアリング部

鮫島　圭

2024/11/26

PE部マスコット
ペペまる

Tomorrow, Together
KDDI
おもしろいほうの未来へ。
au

自己紹介

■ 鮫島　圭（さめしま　けい） ※濁点付きません

● 経歴

- 2010年4月　　KDDI株式会社　入社
- ～2015年10月　　サーバ系設備の運用保守業務（輪番あり）
- ～2016年9月　　RADUIS開発（ProjectLeader(PL)業務メインで、中身はよくわからなかった）
- ～現在　　DNSの開発・保守業務（権威DNSがメイン、やっぱりPL業務がメイン）
ちょっとだけNTP

● 所属

- エンジニアリング推進本部 プラットフォームエンジニアリング部

● 趣味

- テニス（かれこれ数年やってません…）
- スマホ（いっぱいあります）

※撮り忘れが5台くらい

前置き

■**本資料に関する内容は発表者個人の見解を示すものであり、KDDIとしての総意を述べるものではありません**

■**技術的な内容にはあまり触れず、運用面での話がメインとなります**

■**初登壇です**

- 最後の登壇は大学院の論文発表（2009？）

■**本資料の作成にあたり、DNSOPS.jp及びDNS技術者様の様々な資料を参考にさせて頂きました**

- 本発表が今後のDNS界隈の発展に少しでも寄与できれば幸いです

BIND

はじめに


BIND
DNSと言えばBINDじゃないの？
名前だけは聞いたことある
あ、お疲れ様です。。。
脆弱性が多すぎる！
ネットで調べると大体これだよね
あの悪名高い・・・
BIND?Unbound?PowerDNS?
いつもパッチ作業やってない？
よくCVEが出てるやつ
え？BIND使ってるんですか？？(笑)
いい加減卒業したい！

はじめに

BIND卒業できた？

KDDIは
BINDほぼ卒業しました！

2018年の話

BIND卒業できてからの

そして現在

BIND卒業やめました！

(一部)また戻ってきました!

えぇぇ...

BINDから移行しようとした（2018年 DNS Summer Dayにて発表）

■ BINDとの決別を決意

- **BIND脆弱性多すぎ！**
- **DNSの台数多すぎ！**
  - 片手で数えきれない脆弱性対応を、体の指で数えられない台数分実施
  - 脆弱性対応中に更に脆弱性が発生することも（実質作業回数減ったね！…とはならない(怒)）

■ 脆弱性発生からバージョンアップまでに攻撃を受けると非常にマズい

- **準備も作業も時間がかかる（作業まで数営業日、終わるまでに1か月弱）**
- **1時間超のサービス影響 ⇒ 総務省報告**

■ ソフトウェアロックからの脱却

- 今後もBINDだけを使い続けるのか？

【参考】DNS Summer Day 2018 「例の有償ソフトでDNS構築してみた」(松本 章)
https://www.dnsops.jp/event/20180627/(掲載用)DNS_Summer_day_1.1.pdf

BINDから移行しようとした（2018年 DNS Summer Dayにて発表）

■ 各種有償DNSプロダクトで構築を実施
- Akamai DNSi CacheServe（旧 Nominum Vantio CacheServe）
- BIG-IP DNS
- XACK DNS
- その他色々
  - それぞれの使用感については、参考資料をご確認ください

■ 各システム毎に（色々ありましたが…）移行が完了し、運用を開始しました
- 本当に色々ありました・・・
- 本当に本当に、色々ありました・・・

無事、BINDを(ほぼ)卒業しました

2018年の話

【参考】DNS Summer Day 2018 「例の有償ソフトでDNS構築してみた」(松本 章)
https://www.dnsops.jp/event/20180627/(掲載用)DNS_Summer_day_1.1.pdf

色々あったけど

脱BINDしたから脆弱性ほぼ出ない！

余計な作業ともオサラバだ！

話はそんなに単純ではない

…とはなりませんでした

えぇぇぇぇ.....

本当に楽になったのか・・・？

■ 『脆弱性対応に伴う作業』は『減りました』が、その他作業が増えました

● バグFIX作業

- ソフトウェア固有のバグや仕様の修正
  - 大体運用開始してからしばらくして判明する（しかも意外と多い！）
  - DNS的にアウトなバグや、仕様的にイケてないもの等（実質脆弱性では・・・？）
- 最初に引き当てた場合、**解決までに膨大な稼働を要します（数ヶ月単位）**
  - 本社への改修要求(数週間～数ヶ月)、WAの検討(運用稼働大)、進捗管理、改修仕様レビュー、システム改修費、システム再設計、試験・作業計画・作業対応、各種社内処理（社内決済、週次進捗説明、各種承認処理）

● 付帯系設備の作業

- 付帯系もバグFIXやバージョンアップが必要
  - そのためにDNSサーバ側を操作しないといけない場合も…

■ バグFIXは『セキュリティパッチのみ適用』というパターンが少ない

● 『このバグはVersion X.X.X 以降で修正されています（する予定です）』

- 大体新機能が追加されていたり、動作仕様が変わっていたりする（再試験 ⇒ **お金！時間！**）

BINDの場合

■ 回数は多いけど、洗練されているから実質低コスト

● バグFIX作業（脆弱性対応作業）

- いつもの検証／いつもの作業承認／いつもの作業
  - フル内製化済みのため実費の発生なし、作業承認も「いつもの」でOK

● 付帯系設備の作業

- BINDは付帯系システムとかないのでなし
  - 他のシステムと連携している場合はその限りではないです

● バグFIXは『セキュリティパッチのみ適用』

- ESV版BINDのセキュリティパッチは新機能が(ほぼ)ないので、検証も最低限でOK
  - 攻撃コードが公開されていたり、DNSの仕組み上の修正があれば試験する

BINDから乗り換えた際に考えられる影響

■ BINDで構築した方がメリットがある場合もある

- **権威キャッシュDNS** 負の遺産
  - 機能の分離は非常に大変（特にPrimaryDNSが制御できない（他社等）場合）
  - 一部有償DNSでも権威キャッシュ機能は実現可能だが、特殊な構成
- **コスト**
  - 台数が多いほど重くのしかかってくる
    - （ライセンス料 + 保守費）× 台数

■ BINDとの応答差分

- **細かい仕様の違い、RFCの解釈の違い、BIND特有の応答**
- **特に、ユーザセカンダリDNSでは応答が変わるのが許容できない**
  - 権威DNSのソフトウェア切替え時には『**全レコード**』の応答差分確認を実施しました

BIND以外のOSSは？

■ BIND以外にもOSSはあるし、そっちの方が脆弱性も少ないのでは？

● 脆弱性の「件数」は少ないけど、「作業回数」にしたらほぼ同じ（運用目線）

- まとめて出しているだけかもしれないですが…
- 脱BINDの流れがあったのは2016～2018頃、その頃は確かにBINDの脆弱性が圧倒的に多かった

● キャッシュDNSに限って言えば、ここ数年はどれも同じくらいの頻度

- 使用者の増加に伴い、脆弱性も発見されやすくなっている？
- 権威DNSについてはBIND以外が脆弱性少なくて良さそうですね

表1. 各ソフトウェアの脆弱性発生**回数**（日付単位）

	2016	2017	2018	2019	2020	2021	2022	2023	2024
BIND	6	5	5	5	3	4	3	3	2
Unbound	0	0	1	2	2	0	2	0	4
NSD	0	0	1	1	1	0	0	0	0
PowerDNS(A)	0	1	2	3	1	1	1	0	0
PowerDNS(R)	0	1	3	1	3	0	2	2	3
KnotDNS	0	0	1	0	0	0	0	0	0
Knot Resolver	0	0	4	2	1	1	1	2	1

※「https://jprs.jp/tech/index.html」より抽出
※**同日に発表された脆弱性を纏めて1回としています**
※発生回数のみでカウントしています（緊急度は考慮外）

とは言っても・・・

- **OSSを使っている以上、脆弱性対応は切っても切れないから導入はちょっと…**
  - **いつまた頻発するか分からない（だってあのBINDだよ？）**
    - OSSでないから脆弱性が出ないわけじゃないけど…
    - 脆弱性が出た時に稼働がなかったらどうする？
    - それまでに攻撃を受けたら？
- **有償ソフトウェアと比較すると、防御機能が弱いから怖い**
  - **一応ない訳ではないけど…**
    - 機能としては貧弱（高い有償ソフトが優秀とも言える）

機能がないなら...

機能の自作＆自動化
すればいいじゃない

今更感

メインの方が短い

前置きが長くなりましたが…

有償だからって必ずしも良いわけじゃない！ってのが言いたかった

BINDから移行しようとした（2018年 DNS Summer [illegible]にて発表）

自動化すれば
ほぼ解決！

■ BINDとの決別を決意

自動化すれば
解決！

- BIND脆弱性多すぎ！
- DNSの台数多すぎ！

自動化すれば
解決！

  - 片手で数えきれない脆弱性[illegible]で数えられない台数分実施
  - 脆弱性対応中に更に脆弱性が発生することも（実質作業回数減ったね！…とはならない(怒)）

■ 脆弱性発生からバージョンアップまでに攻撃を受けると非常にマズい

- 準備も作業も時間がかかる（作業まで数営業日、**終わるまでに3週間**）
- 1時間超のサービス影響 ⇒ 総務省報告

自動化すれば
リスク軽減

■ ソフトウェアロックからの脱却

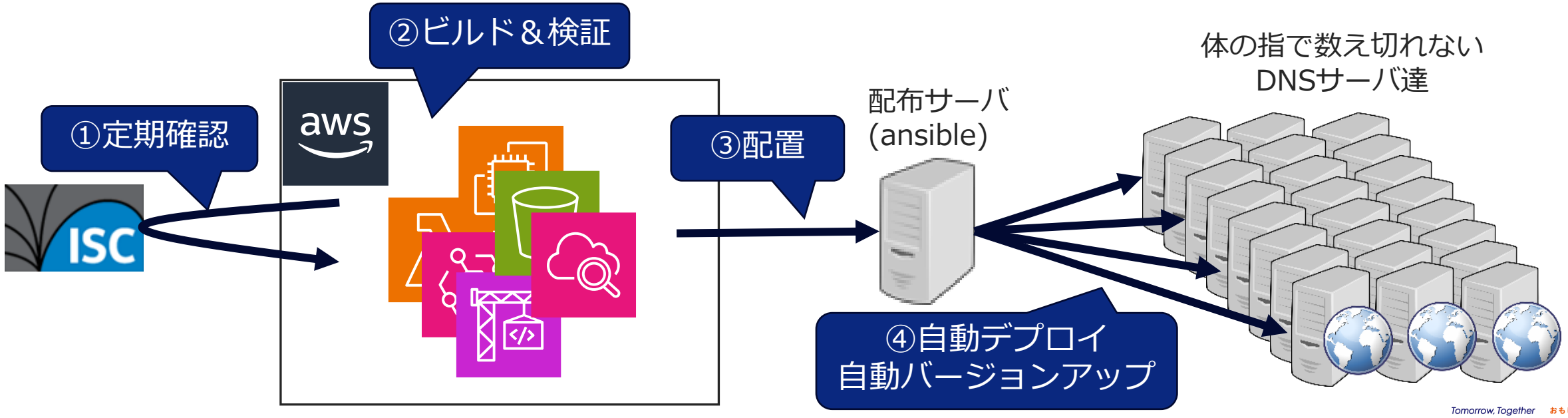
  - 今後もBINDだけを使い続けるのか？

実績はできた

【参考】DNS Summer Day 2018 「例の有償ソフトでDNS構築してみた」(松本 章)
https://www.dnsops.jp/event/20180627/(掲載用)DNS_Summer_day_1.1.pdf

自動化による自動バージョンアップ機能の実装

- 定期的に最新バージョンを確認し、更新があった場合は自動でビルド＆検証し配布サーバまで設置
- スケジュール・対象サーバを設定すれば自動的にデプロイ＆バージョンアップ
- 脆弱性対応期間の大幅な短縮が可能に
  - 緊急時はパッチ公開後即時対応も可能



その他自動化とか 外部ツールとか

■ ランダムサブドメイン攻撃に対しての防御機能が全くなかった時代

– fetches-per-server ／ fetches-per-domain

- **ログを定期的に走査し、攻撃と思われるログ傾向を検出したら自動的にブロックする機能を実装**
  - 攻撃が収まったタイミングで自動解除
  - 稀に誤検知はあったものの、改修やパラメータ修正を重ね概ね良好に動作
    – 詳細な動きを記載しようと思ったが、この余白はそれを書くには狭すぎる

■ その他、細かい機能の実装（外部ツール含む）

- BINDプロセスの監視 ⇒ プロセス落ちたら自動起動
- 一定QPSを超過したらネットワーク側でmitigation（NW担当と連携）
- 統計機能（named statsの繰り返し取得+集計）

これまでの自動化、外部ツール作成を通して

■ 弱点や足りない機能は自分達なりの方法で補完することで実用レベルにしてきた

- 「プロセスの自動再起動する機能ないから、落ちたらそのままでいいやー」とはならないですよね？
- あとはやる気次第

■ 『自動化』という単語がより一般的に言われるようになり、以前よりも敷居は確実に下がっているはず

- 自動化やツール作成により、一部BINDでも問題ないと判断
  - 脆弱性対応を自動化しても、ゼロデイアタックは相変わらず怖いですが

■ 足りない機能の補完方法は共有したいですね

- 集約してBINDベストプラクティス的なものを作っても良いかも？
- 会社としては競合でも、技術者としては仲間（だと思っています…）

ReBIND

【参考】BIND辞められない理由 Q&A
https://www.dnsops.jp/event/20160624/BIND_QandA_配布用.pdf

BINDという選択肢もありじゃないでしょうか

- **以前程脆弱性祭りもなくなった**
  - キャッシュに関しては他のOSSも同じ状況（権威は(扱えるなら)他のOSSでも良さそう）
  - 脆弱性があった場合に備えて複数OSSを使用する？⇒運用が倍大変になる！
    - 緊急用に別のDNSソフトウェアを置いておく、というのはどうでしょう（対処までの数日凌げればOK！）
      - やってました、役に立ってしまいました（数ヶ月凌ぎましたが…）
- **既存システムがBINDの場合、大規模になる程、変更は大変**
  - 「BIND固有の機能」「脆弱性対応の工数 << 作り直す工数」「学習・教育コスト」、etc...
    - 「運用者」としては重要なファクター
- **「BIND」vs「その他OSS」vs「有償ソフトウェア」**
  - どんな製品を導入しても「これで全て解決！」とはなりません…
    - BIND　・・・・・・・・・脆弱性がー、防御力がー
    - その他OSS　・・・・ドキュメントがー、機能追加がー、（脆弱性がー）
    - 有償ソフト　・・・・パッチ適用がー、価格がー
- **要件に合わせた適切なソフトウェア選定を！（何だかんだBINDは色々楽です）**

Tomorrow, Together

KDDI

「つなぐチカラ」を進化させ、
誰もが思いを実現できる社会をつくる。

KDDI VISION 2030