東海大学大学院 2012 年度 修士論文

DNS キャッシュ保持に関する脆弱性
“ghostdomain”の研究

Research on "ghost domain" a vulnerability of the DNS cache

指導教員 菊池 浩明 教授

東海大学大学院工学研究科情報理工学専攻

1BDRM001 有水智大

目次

第１章　　序論

1.1　背景

インターネットの一般家庭への普及が進み,ネットワークに接続できるユーザーが増加している.それに伴いインターネットセキュリティ意識の低いユーザーも増加し,不正な目的を持つホストの標的にされ被害が深刻化している.悪意あるホストは,ドメインネーム（DNS）をフィッシングサーバ,ボットネット指揮統制,およびマルウェア配布など様々な目的に不正使用する.これらの活動を防ぐために行われている対策は,上位 DNS サーバから悪意のある領域を削除することである.このようなドメイン名の差し押さえは,親ゾーンから委任情報(NS レコード)を削除,あるいは強制的に変更するという方法がとられる場合がある. たとえば,phishing.com という不正なサイトがあった場合,com を管理しているレジストリが phishing.com の NS レコードを com ゾーンから削除することでアクセス不可能にしたり,別の NS レコードを設定することで任意のユーザーが phishing.com にアクセスした時に別のサーバへと転送したりする. com ゾーンからの削除ではなく別の NS レコードが設定された例として,megaupload.com の事例があげられる. megaupload.com に関して WHOIS で出力される権威 DNS サーバ一覧と,実際に.com ゾーンに設定されている権威 DNS サーバの一覧が異なっている. WHOIS では Megaupload Limited による NS が記載されているが,実際に.com ゾーンで NS として指定されている cirfu.net の登録者は Federal Bureau of Investigation(FBI)である.

しかし,上位 DNS サーバから取り除かれた後も,悪意のあるドメインネームが稼働し続けることを可能にする DNS の脆弱性が Haixin Duan ら[1]によって報告された.彼らは 19,045 のパブリック DNS サーバを用いた実験で,ドメインネームが無効にされ,有効期間 TTL の 1 週間後も 70%以上のサーバがまだ名前解決出来ることを示した.この脆弱性によりキャッシュ DNS サーバにおいて名前解決可能な状態にさせられ続けてしまうドメイン名を示す用語として「幽霊ドメイン名」を使用する.

1.2　目的

そこで,本研究では,現在の主なパブリック DNS サーバに幽霊ドメイン名脆弱性が存在するか調査を行い,脆弱性の危険性を明らかにする.また脆弱性の DNS ソフトウェア毎の調査や地理的分布,幽霊ドメインの生存期間の調査を行い,パブリック DNS が行っている対策や傾向を考察する.

1.3　論文構成

本論文の構成は次の通りである. 第２章では, 研究に関連する技術について述べる. 第３章では, 関連する研究について述べる. 第４章では, 幽霊ドメイン名攻撃を再現し, パブリック DNS サーバの現在の幽霊ドメインへの対応状況を明らかにするために実験を行う. 第５章では, 実験結果をもとに考察を行う. 最後に第６章で結論と今後の課題について述べる.

第２章　　要素技術

2.1　DNS

DNS はインターネットを構成する非常に重要なプロトコルである.インターネットを利用する上で最も使用され,なくてはならない存在である.DNS とはドメイン名と IP アドレスの対応システムである.ブラウザやメーラなどで指定したサーバの IP アドレスを知るために使われる.DNS の概略を図 2.1 に示す.インターネット利用者が www.example.com のアドレスを知りたいとき,DNS に問い合わせる.DNS はまず一番親である root サーバにそれが管理する com のアドレスを聞く.次に com にそれが管理する example のアドレスを聞く.最後に example に www のアドレスを聞き,利用者に送信する.このように DNS ではドメイン名は分散されて管理されている.

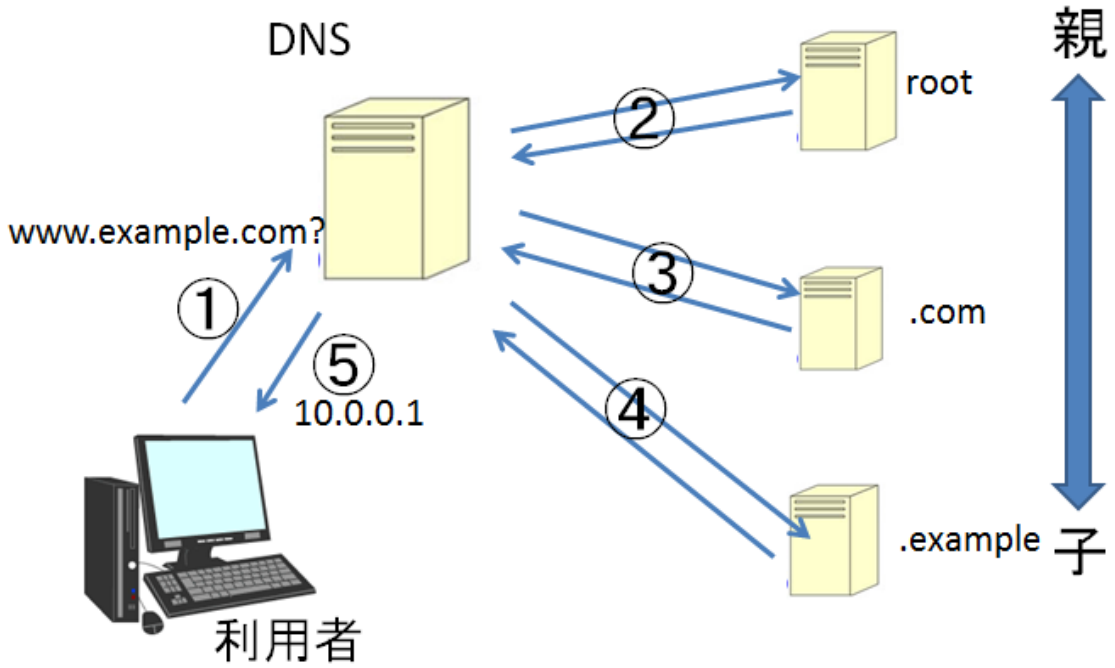


図 2.1 DNS

2.2　DNSSEC

DNSSEC は DNS のクエリを偽造して本来とは違うアドレスにアクセスさせるキャッシュポイズニング攻撃など,DNS を利用した様々な攻撃に有効な仕組みである.DNSSEC は電子署名の仕組みを応用して,DNS キャッシュサーバが問い合わせによって得た応答が,問い合わせた権威 DNS サーバからの応答かどうか,パケットが改ざんされていないかなどを検証することができる.

DNSSEC ではこの仕組みを公開鍵暗号方式と電子署名を組み合わせて実現している.図 2.2 に DNSSEC の概略を示す.キャッシュ DNS は権威 DNS が公開している公開鍵を事前に取得し,保存しておく.権威 DNS に問い合わせるとき,キャッシュ DNS は問い合わせのレコードをハッシュ化し,控えておく.その後レコードの問い合わせを行い,それを受け取った権威 DNS はそのレコードを同じくハッシュ化する.さらにそのハッシュ値を秘密鍵で暗号化し,署名としてレコードに付与してキャッシュ DNS に応答する.キャッシュ DNS は受け取った署名を公開鍵で復号し,最初にレコードをハッシュ化したものと復号結果が一致すればその応答を信用する.

しかし,この方法で検証を行うには,権威 DNS の公開鍵が確かにその権威 DNS のものであると確証できることが前提となる.DNSSEC ではこの公開鍵の確証を“信頼の連鎖“という仕組みにより保証している.

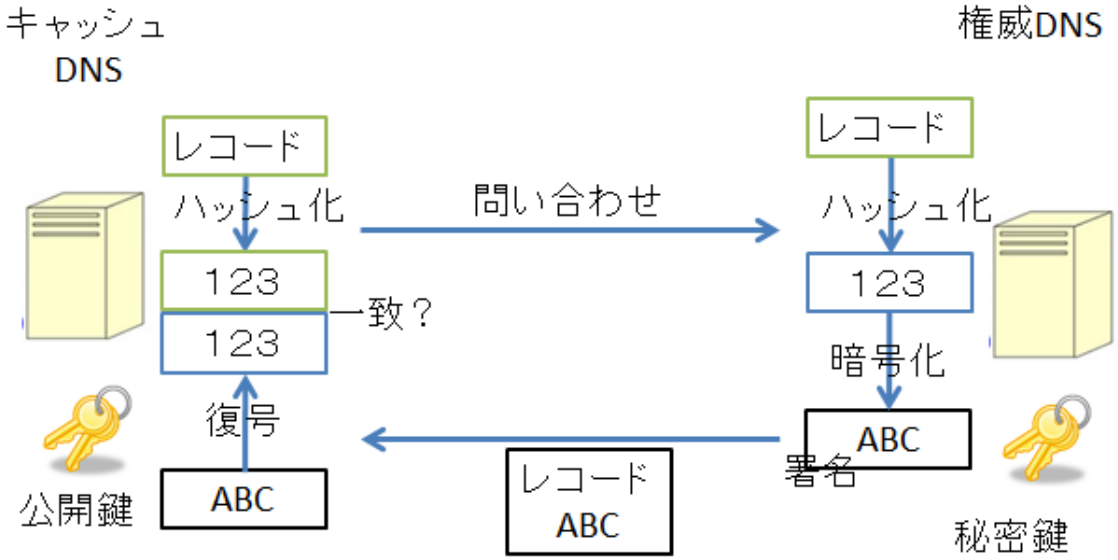


図 2.2　DNSSEC

信頼の連鎖の概略を図 2.3 に示す.各 DNS サーバは,自分の管理するゾーンの公開鍵を事前に取得する.キャッシュ DNS は root の公開鍵をセキュアな方法で事前に取得する.キャッシュ DNS はあるドメインの名前解決を行う時,まず root に問い合わせる.事前に root の公開鍵を取得しているため,ここでは信頼された応答が行える.この信頼された応答で,次に問い合わせを行う DNS の公開鍵を root に署名付きで送信してもらう.このようにして次に問い合わせる DNS の信頼できる公開鍵を取得することができる.次にその取得した公開鍵で再び問い合わせ先の DNS に管理するゾーンの公開鍵を署名付きで送信してもらう.とこのように信頼を連鎖させることで最終的な目的地の DNS の公開鍵をもセキュアに取得することができる.

DNSSEC はセキュリティが向上する利点があるが,導入には技術的課題がいくつかある.ひとつは DNS 応答パケットサイズの増加である.DNS はパケットサイズが 512byte までの UDP で通信を行うが,DNSSEC はそれを超えてしまうため,扱えるパケットサイズを増やすか,TCP で通信を行わなければならない.また,パケットサイズ増加によるサーバ負荷の増加も課題である.

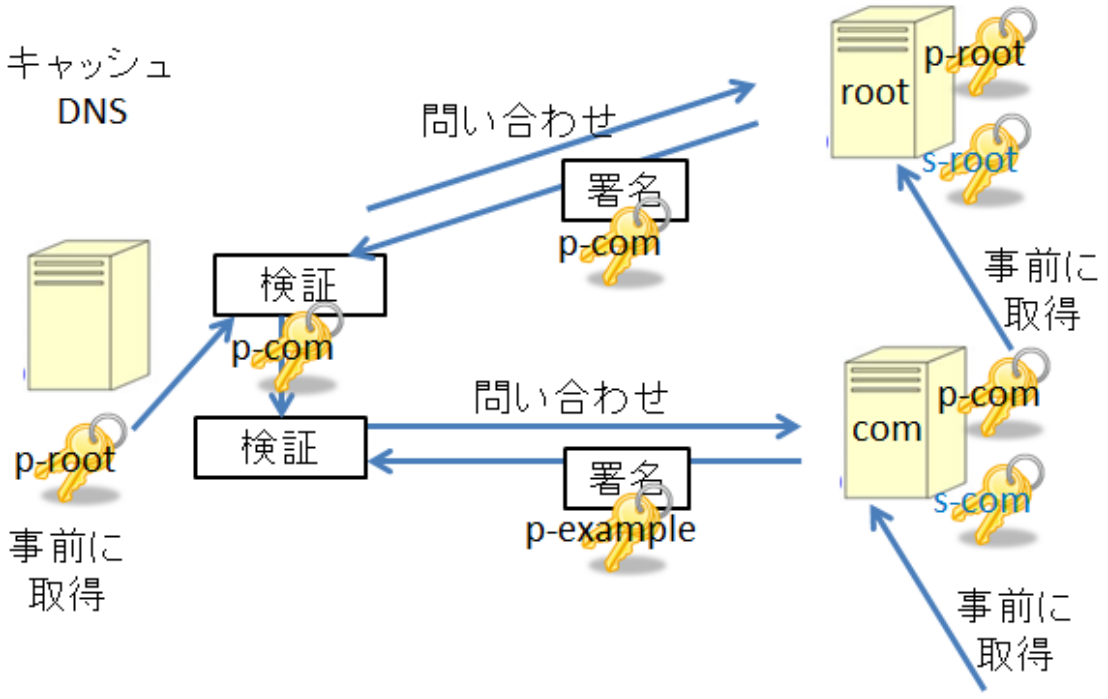


図 2.3 信頼の連鎖

第３章　　関連研究

3.1　正規 DNS 応答トラフィックからの DNS キャッシュポイズニング攻撃検知

DNS キャッシュを悪用した攻撃のひとつに,DNS キャッシュポイズニング攻撃がある.この攻撃の中でも高度な攻撃と呼ばれる Kaminsky 攻撃[5]があり,武蔵らはこの攻撃を検知する手法を提案しており,実際に攻撃を観測したデータから評価を行っている[4].

3.1.1　背景

フィッシングやスパム,DDOS 攻撃はボットネットワークを構成しているため,これらを検知することはボットの発見率を上げるために重要である.近年,フィッシングに DNS キャッシュポイズニング攻撃が使われ,Kaminsky 攻撃は最も高度な攻撃と言われている.

3.1.2　キャッシュポイズニング攻撃

キャッシュポイズニング攻撃はキャッシュ DNS の問い合わせに対する応答を偽造し,本来とは違うドメインと IP を関連付けるキャッシュをキャッシュ DNS に残す攻撃である.攻撃手順を図 3.1 に示す.ここでは,攻撃者は www.example.com にアクセスしにくるユーザーを b.b.b.b に誘導したい.１．攻撃者は攻撃対象キャッシュ DNS サーバを介して www.example.com A レコード検索を行う.２．キャッシュ DNS サーバは権威 DNS サーバにクエリ偽造防止のための ID を付与して A レコードの問い合わせを行う.３.権威 DNS からキャッシュ DNS へ応答が帰ってくる前に攻撃者は ID を総当たりに偽造したクエリをキャッシュ DNS に送信する.ID が一致すればキャッシュ DNS はそれを正規の応答として受理し,www.example.com を b.b.b.b と関連付ける.４.遅れて権威 DNS が応答を送るが,既に応答を受け取っているキャッシュ DNS はその応答を破棄する.

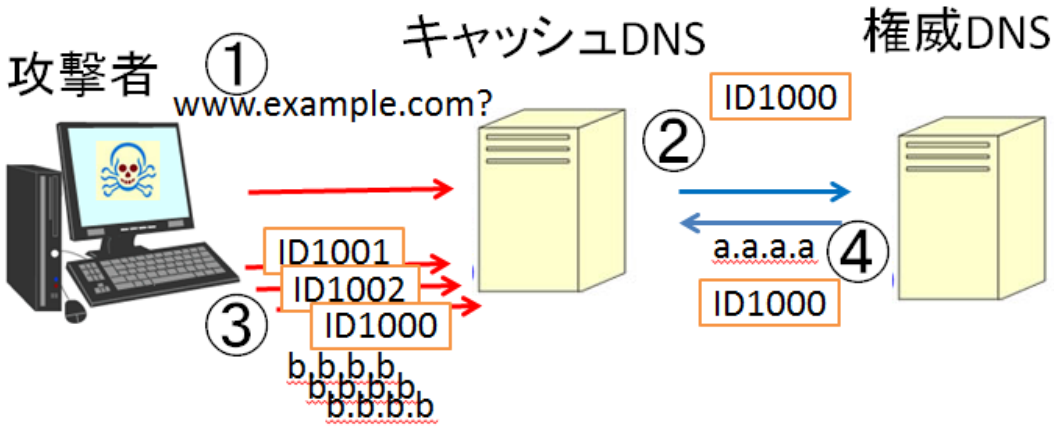


図 3.1　キャッシュポイズニング攻撃

3.1.3　Kaminsky 攻撃

Kaminsky 攻撃はキャッシュポイズニング攻撃を高度にしたもので,従来の攻撃では攻撃に失敗した時,ネガティブキャッシュが残ってしまうための TTL 期間内は再び攻撃することができなかった.しかし Kaminsky 攻撃では検索するドメイン名をランダムに変更することで攻撃に失敗しても連続して攻撃を続けることができる.

攻撃手順を図 3.2 に示す.ここでは,攻撃者は www.example.com にアクセスしにくるユーザーを b.b.b.b に誘導したい.1．攻撃者は攻撃対象キャッシュ DNS サーバを介してランダムに生成したドメイン xxx.example.com A レコード検索を行う.2．キャッシュ DNS サーバは権威 DNS サーバにクエリ偽造防止のための ID を付与して A レコードの問い合わせを行う.3．権威 DNS からキャッシュ DNS へ応答が帰ってくる前に攻撃者は ID を総当たりに偽造したうえに,www.example.com と b.b.b.b を関連付けたクエリをキャッシュ DNS に送信する.ID が一致すればキャッシュ DNS はそれを正規の応答として受理し,www.example.com を b.b.b.b と関連付ける.4．遅れて権威 DNS が該当ドメインなしと応答を送るが,既に応答を受け取っているキャッシュ DNS はその応答を破棄する.このように存在しないドメイン名を使用することで,ネガティブキャッシュにかかることなく攻撃を行える.

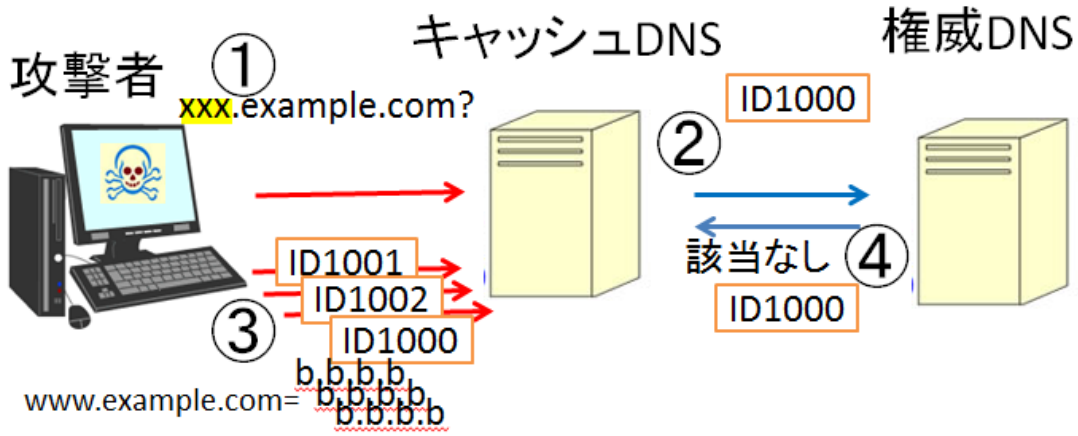


図 3.2　Kaminsky 攻撃

3.1.4　攻撃観測

筆者らは 2010 年 1 月 1 日から 12 月 31 日の間,熊本大キャンパスネットワーク内トップドメインの DNS を観測し,二つの方法で 5 回の Kaminsky 攻撃を検出した.

1)DNS クエリパケットトラフィックの IP アドレスによるエントロピーの急速な低下時,またユニーク DNS クエリキーワードによるエントロピーの急増時

2)観測クエリキーワードと次のキーワード間の Damerau-Levenshtein 距離を用いた検出法における 1～40 の閾値範囲

3.1.5　武蔵らの攻撃検知手法[4]

DNS クエリトラフィックエントロピーは以下のように定義される.

$$H(X) = -\sum_{i \in X} P(i) log_2 P(i)$$

$$P(i) = \frac{freq(i)}{(\sum_j freq(j))}$$

i はユニーク IP,j はユニーク DNS クエリ,freq(i)は著者の過去の研究[6]のスクリプトによって概算される.Kaminsky 攻撃を検知するための閾値を一日当たり 40000 パケットの上位 10 位のユニークソース IP アドレス,もしくは DNS クエリキーワードとする.Damerau-Levenshtein 距離は以下のように定義される.

$$LD[x,y] = \min(LD[x-1][y]+1, LD[x][y-1]+1, LD[x-1][y-1]+cost)$$

x,y は文字列 X,Y の長さであり,X はある点での FQDN であり,Y はその次に来た FQDN である.例えば,X = “a001.example.com”　Y = “a002.example.com” の場合は距離は 1 となる.

3.1.6　結果

図 3.3 はエントロピーによる Kaminsky 攻撃検知の結果を示しており,文献[4]から引用している.2010 年 1 月 25 日～29 日にかけて攻撃の検知に成功している.Kaminsky 攻撃では権威 DNS からの応答がかえってくる前に大量の偽造クエリを送る必要がある,そのため攻撃をしている IP は大量のクエリを送信することになり,ユニーク IP アドレスベースのエントロピーは低下する.その一方で,存在しないドメインに対して A レコード検索クエリを送信する必要があるため,攻撃者はクエリごとにランダムな文字列を生成する.その結果ユニーク DNS クエリキーワードベースのエントロピーは上昇する.

図 3.4 は Damerau-Levenshtein 距離の年間スコアを示しており,文献[4]から引用している.この結果は誤検知が多かった検出方法を改良したものである.エントロピーによる Kaminsky 攻撃検知と同じように,1 月あった攻撃の検知に成功しているが,改良前のものは 3 件の誤検知があった.その理由として,攻撃者は大文字・小文字を多用して架空ドメインを作成していることがあげられ,大文字小文字を区別せずに計算することで誤検知を減少することができている.

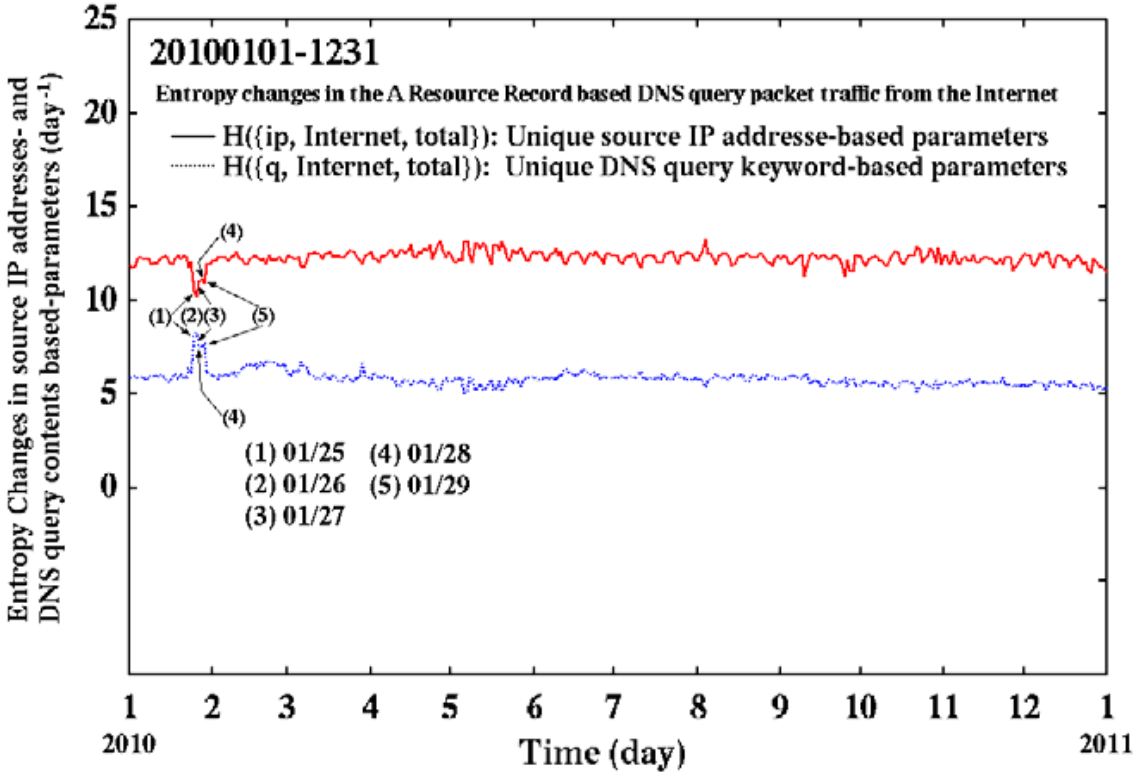


図 3.3 エントロピーによる Kaminsky 攻撃検知の結果(文献[4]より引用)

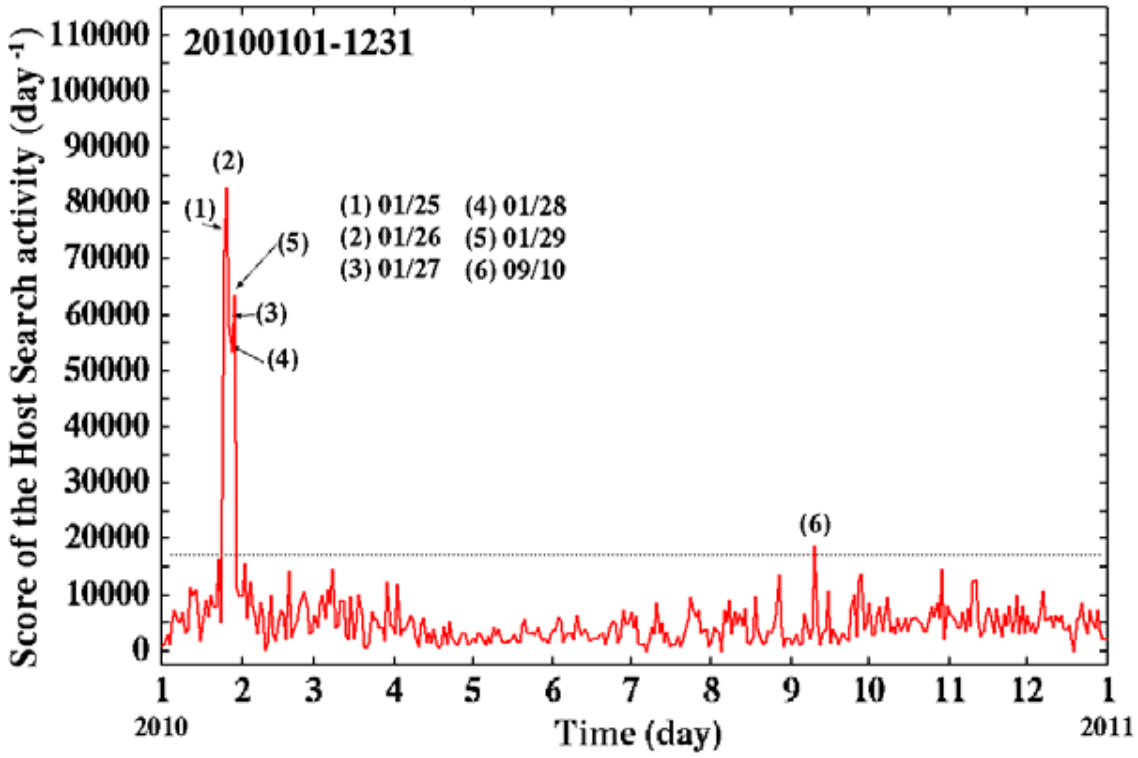


図 3.4 Damerau-Levenshtein 距離の年間スコア(文献[4]より引用)

3.2　複数のキャッシュ DNS サーバーを利用した安全な名前解決方法

キャッシュポイズニング攻撃やスプーフィングを防ぐ手段として 2.2 節で述べた DNSSEC があるが,各 DNS にまだ浸透していない,パケットサイズの増加などの問題がある.堀らはクライアントサイドで複数のキャッシュ DNS サーバを用いた名前解決や,パケットの監視を行うことで,サーバやユーザーの負担なく不正を防止する手法を提案している.[7]

3.2.1　背景

DNS は欠かすことの出来ない重要なインフラである.しかし,DNS サーバ内のキャッシュを改ざんしたり,名前解決応答を偽造することでクライアントを不正なサーバに誘導することができる.DNS を利用した攻撃として DNS スプーフィング（なりすまし）攻撃や DNS キャッシュ汚染攻撃などがあげられる.

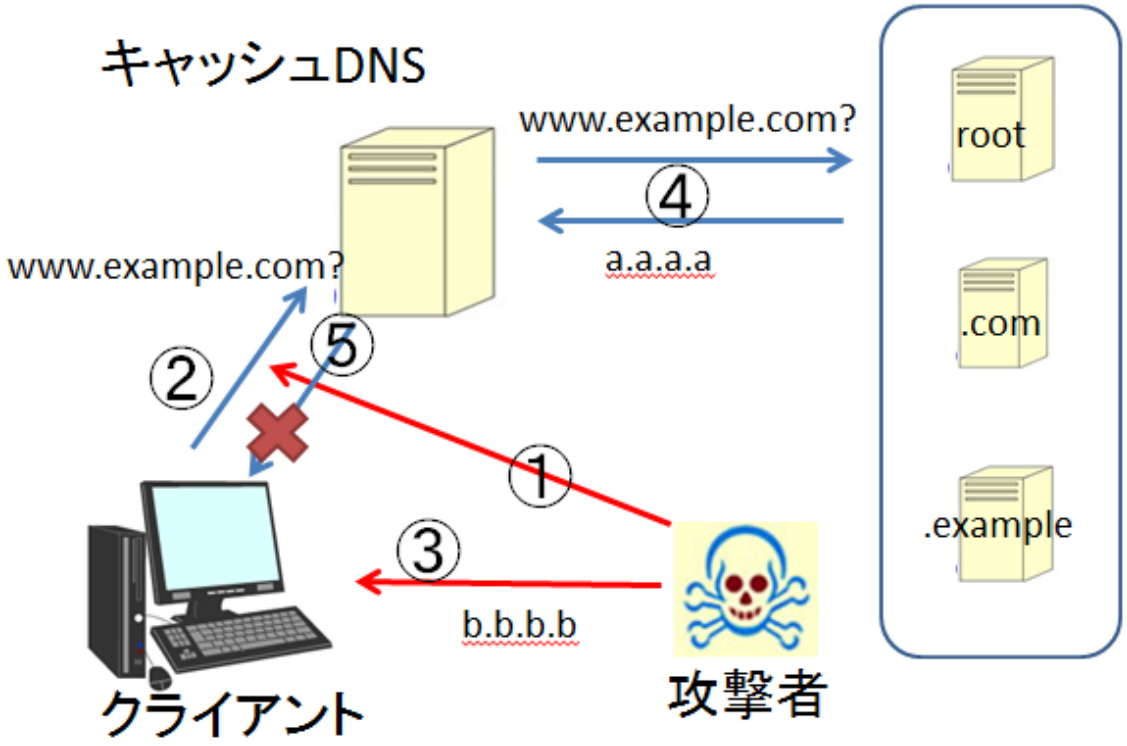


図 3.5 DNS スプーフィング攻撃

3.2.2 対策方法

一般的な対策方法としてはDNSSEC（DNS Security Extensions）がある.DNSサーバが送信するデータに署名を付与しそれを検証することで信頼性を保証する.しかし導入に時間がかかり,現在浸透に至っていないのが現状である.実際に行われている対策としては,DNS サーバソフトのバグフィックス,ゾーンデータの正しい設定などが行われる.しかしこれらを行っていないDNSサーバは数多く存在する.関連研究ではDNS回答信頼度算出システムによる対策を提案している.DNS に信頼度を計測するための問い合わせをネームサーバに行う機能を実装している.しかし,クライアントのみで行うことはできないため,実装が容易ではない.

3.2.3 堀らの検知手法[7]

名前解決をする際,複数のDNSに対して名前解決要求を送信して比較する.また,送信する名前解決要求と受診する応答の数を確認する.C言語を用いて実装しており,パケットキャプチャーにはlibpcapライブラリを使用している.

複数のDNSによる名前解決する手法を図3.6に示す.クライアントはあらかじめDNSIPのリストを用意しておく.クライアントは閾値kをキャッシュDNSの数nを考慮して任意に定める.ある名前解決をしたいとき,DNSIPリストの全てのDNSに対して名前解決要求を送る.応答を受け取ったら,ユニークな返答毎に総和を求め,k以上のDNS応答を採用する.

DNSパケット監視機能を図3.7に示す.基準値s=0と定め,クライアントが名前解決要求を行うたびにsをインクリメントする.そして応答を受け取るたびにsをデクリメントする.名前解決が終了したとき,s<0となっていた場合警告を出す.ここでは,攻撃者はDNSスプーフィング攻撃を行おうとしており,クライアントとDNSの通信を監視しているとする.クライアントは www.example.com の名前解決要求を送信し,s=1 となる.攻撃者はこれを監視し,DNSから応答がかえる前にクライアントに対して偽の応答b.b.b.bを送信し,s=0となる.その後DNSは正規の応答a.a.a.aを送信し,s=－1となる.このように要求と応答の数が合わないような攻撃の検知ができる.

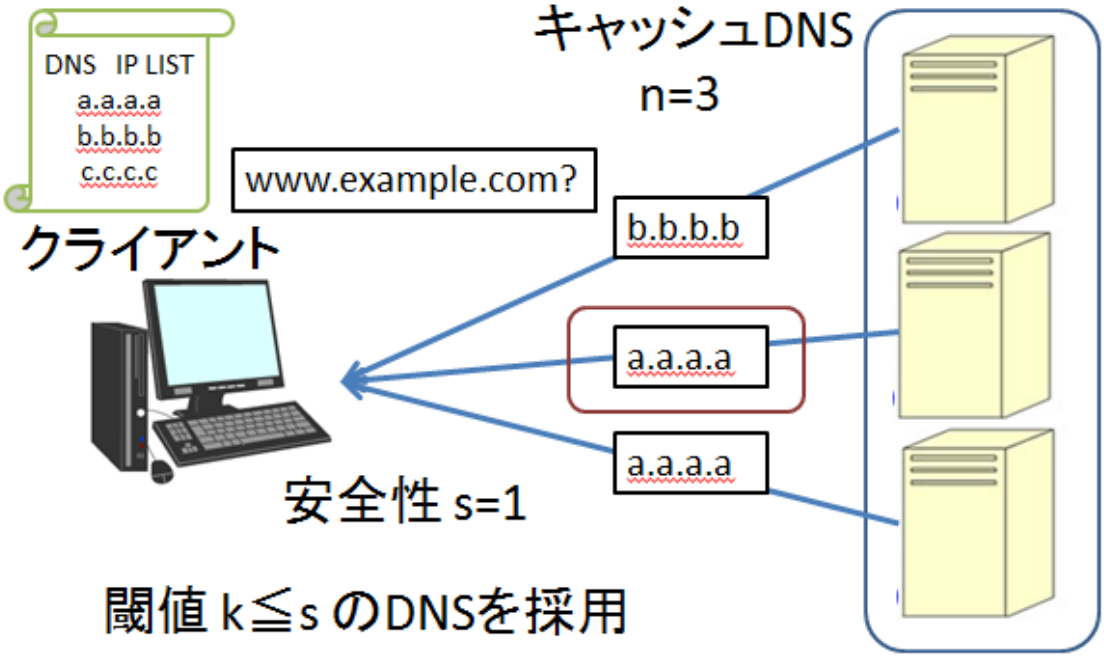


図 3.6 名前解決結果比較機能

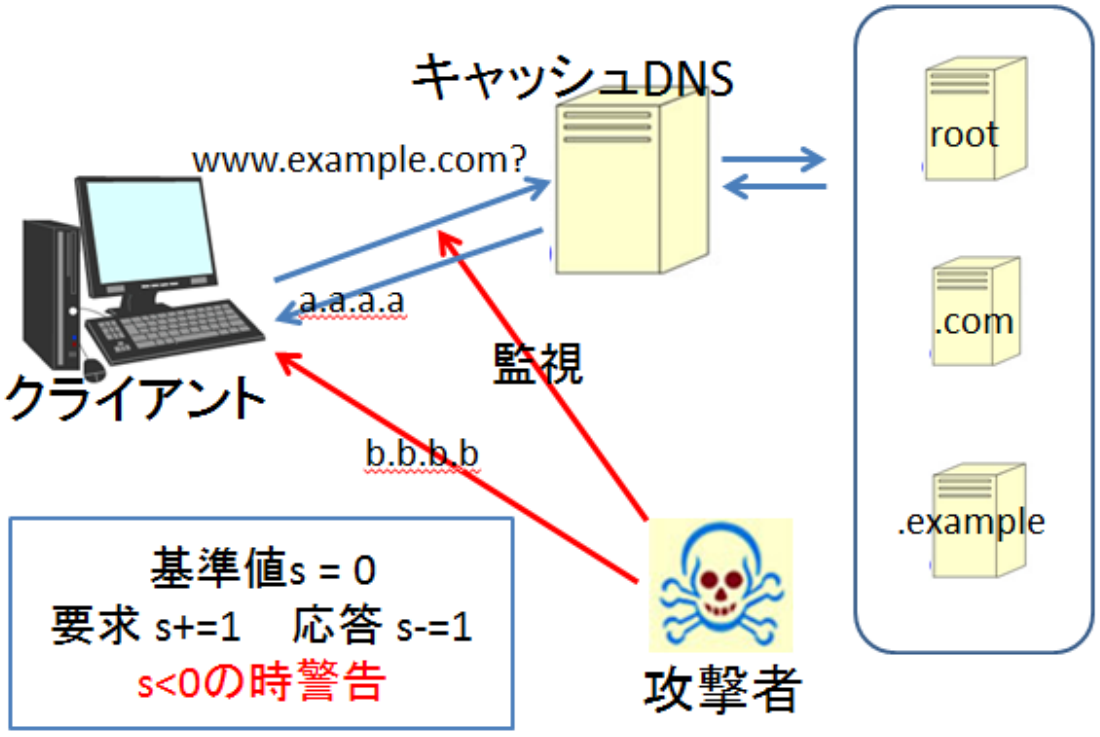


図 3.7 DNS パケット監視機能

3.2.4 評価

名前解決要求が 1 個 2 個 3 個 4 個 5 個の時それぞれについて処理時間を計測し,通常の名前解決と提案方式を利用した場合を比較している.名前解決するドメインには実験前にあらかじめ問い合わせを行いキャッシュを保存させた状態で行っている.名前解決要求が１～５個の場合それぞれについて１００回行い平均を算出している.処理時間は最大でも通常の処理時間の 2.72 倍と十分実用に耐える速度である.

表 3.1　処理時間の計測（文献[7]より引用）

DNSCacheServer の数	平均処理時間(秒)	通常の処理時間を 1 としたときの倍率
1(通常の名前解決)	0.04911	1.00
2	0.07364	1.49
3	0.09855	2.01
4	0.11886	2.42
5	0.13369	2.72

3.2.5 考察

提案方式により DNS キャッシュ汚染や DNS スプーフィングに対応できるが,以下のような攻撃を防ぐことができない.

・設定した閾値以上の DNS サーバが一様に改ざんされている場合

・攻撃者がクライアントから送信される名前解決をフィルタリングした場合

しかし,アプリケーションへの組み込みが可能で負荷も小さいために有用であるといえる.

第４章　　実験

4.1　実験１

4.1.1　実験概要

脆弱性が発表されてから約一年が経過した 2013 年１月現在において,パブリック DNS サーバにどれほどの幽霊ドメイン名脆弱性が残っているかを明らかにするため,幽霊ドメイン名攻撃を再現する.二台のサーバを用いて１つを権威 DNS サーバとし,もう１つの DNS サーバへドメインの委任を行う.調査対象パブリック DNS に委任先のゾーンのキャッシュを残した後,権威 DNS サーバから委任情報を削除し,委任先ゾーンのネームサーバ名を書き換え,調査対象パブリック DNS に残したキャッシュの TTL 値を延長させる.延長しているか確かめるために,ホスト名で A レコード検索を行い,IP アドレスを引くことができれば TTL 値が延長していると判定する.実験で使用したドメインの関係を図 4.1 に示す.

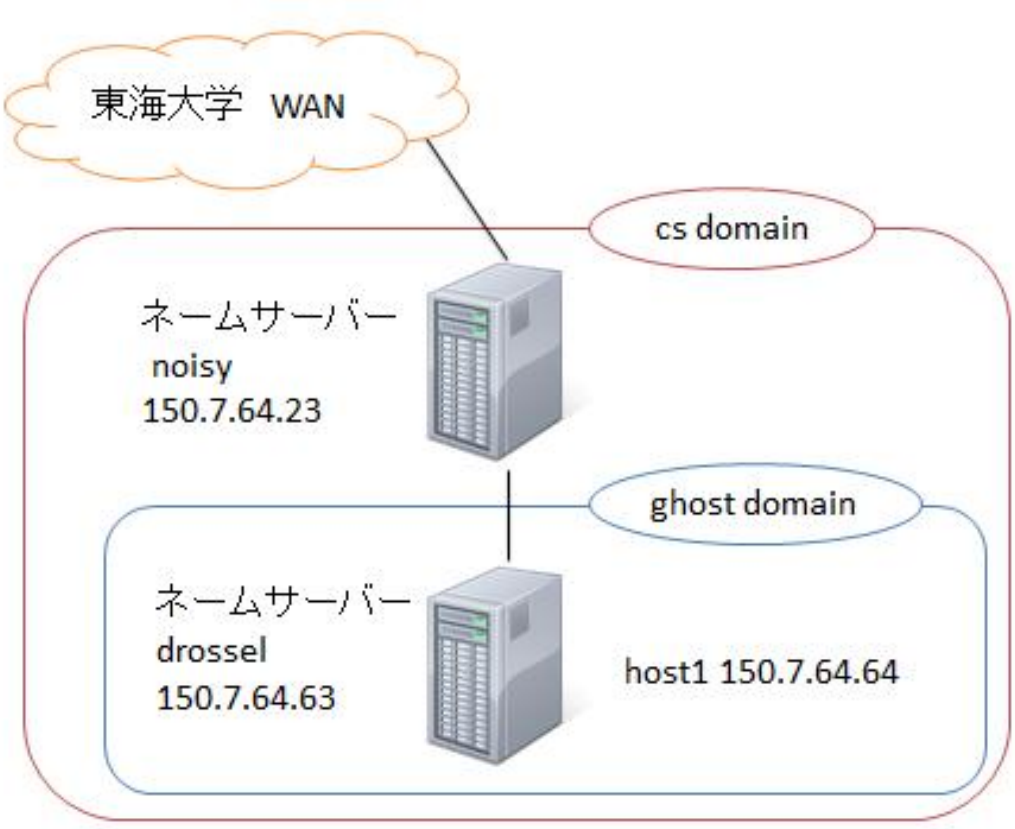
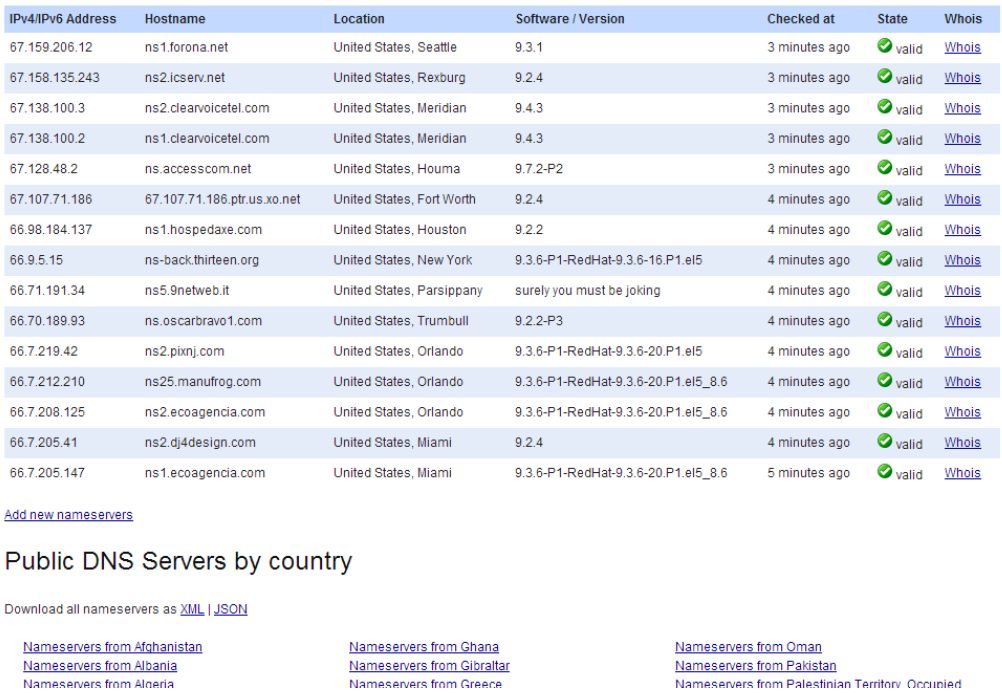


図.4.1　実験に使用したドメインの関係

4.1.2 実験対象

調査するパブリック DNS には,public-dns.tk[3]が公開しているパブリック DNS のデータベースを用いた.2972 個のアドレスから,A レコード問い合わせに対して timeout するものを除いた,2791 件について調査した.public-dns.tk を図 4.2 に示す.2013 年 1 月現在,3386 件 176 か国のパブリック DNS サーバが公開されており,XML,JSON 形式で利用することができる.データの一部を表 4.2 に示す.今回の実験では IPv4 について調査した.



IPv4/IPv6 Address	Hostname	Location	Software / Version	Checked at	State	Whois
67.159.206.12	ns1.forona.net	United States, Seattle	9.3.1	3 minutes ago	valid	Whois
67.158.135.243	ns2.icserv.net	United States, Rexburg	9.2.4	3 minutes ago	valid	Whois
67.138.100.3	ns2.clearvoicetel.com	United States, Meridian	9.4.3	3 minutes ago	valid	Whois
67.138.100.2	ns1.clearvoicetel.com	United States, Meridian	9.4.3	3 minutes ago	valid	Whois
67.128.48.2	ns.accesscom.net	United States, Houma	9.7.2-P2	3 minutes ago	valid	Whois
67.107.71.186	67.107.71.186.ptr.us.xo.net	United States, Fort Worth	9.2.4	4 minutes ago	valid	Whois
66.98.184.137	ns1.hospedaxe.com	United States, Houston	9.2.2	4 minutes ago	valid	Whois
66.9.5.15	ns-back.thirteen.org	United States, New York	9.3.6-P1-RedHat-9.3.6-16.P1.el5	4 minutes ago	valid	Whois
66.71.191.34	ns5.9netweb.it	United States, Parsippany	surely you must be joking	4 minutes ago	valid	Whois
66.70.189.93	ns.oscarbravo1.com	United States, Trumbull	9.2.2-P3	4 minutes ago	valid	Whois
66.7.219.42	ns2.pixnj.com	United States, Orlando	9.3.6-P1-RedHat-9.3.6-20.P1.el5	4 minutes ago	valid	Whois
66.7.212.210	ns25.manufrog.com	United States, Orlando	9.3.6-P1-RedHat-9.3.6-20.P1.el5_8.6	4 minutes ago	valid	Whois
66.7.208.125	ns2.ecoagencia.com	United States, Orlando	9.3.6-P1-RedHat-9.3.6-20.P1.el5_8.6	4 minutes ago	valid	Whois
66.7.205.41	ns2.dj4design.com	United States, Miami	9.2.4	4 minutes ago	valid	Whois
66.7.205.147	ns1.ecoagencia.com	United States, Miami	9.3.6-P1-RedHat-9.3.6-20.P1.el5_8.6	5 minutes ago	valid	Whois

図 4.2 public-dns.tk ウェブページ（文献[3]より引用）

4.1.3 実験環境

研究室内にある 2 台の DNS サーバを用いて実験を行った.2 台の詳細を表 4.1 に示す.drossel の権威 DNS を noisy に設定した.

表 4.1: 実験機材詳細

DNS 名	型名	スペック	OS
noisy	Dell PowerEdge2650	Xeon 2.3GHz 1GB	Linux Redhat 9
drossel	Dell PowerEdge410	Xeon 2.4GHz 4GB	Cent OS 5.3

表 4.2 パブリック DNS データベースの一部（文献[3]より引用）

country-id	id	ip	name	version
AF	9235	210.80.58.66	cache300.ns.uu.net	
AF	9236	210.80.58.67	cache301.ns.uu.net	
AR	2129	200.69.193.1	dns1.iplanisp.com	9.7.3
AT	11457	62.2.100.5	dns1.vtz.net	
AT	7062	80.120.17.70	res2.a1.net	
AT	2642	80.243.161.1		
AT	2764	80.243.162.194	ns2.itandtel.at	
AT	10107	193.33.114.2	ns01.anexia.at	yes, it is BIND
AT	9611	193.109.140.11	ns2.westnet.at	9.7.6-P4
AT	9619	193.111.47.53	ns.oevag.at	OEVAG
AT	13133	193.170.204.18	argus.htl-hl.ac.at	9.6-ESV-R4
AT	5051	193.171.87.249	nsrv1.unileoben.ac.at	
AT	4885	193.171.87.250	nsrv2.unileoben.ac.at	
AT	10002	193.186.16.222	ns.salomon.at	9.6-ESV-R1
AT	13450	194.24.128.100	ns1.orange.at	
AT	13451	194.24.128.102	ns2.one.at	
AT	13407	195.34.133.10	ns1.chello.at	
AT	13408	195.34.133.11	time.chello.at	
AT	10305	195.78.53.25	ns1.csc.at	CSC Austria GmbH
AT	10940	212.24.98.97	ns4.networkorganization.org	9.8.1-P1

4.1.4 実験方法

図4.3,4.4に実験1の流れを示す.今回はゾーンTTL値は3600に設定した.noisy,drosselのゾーン設定は以下のようになっている.

drossel			noisy		
NS ns01.ghost			NS noisy		
ns01.ghost	A	150.7.64.63	drossel	A	150.7.64.63
host1	A	150.7.64.1	ghost	NS	drossel

実験は以下の手順で行った.

1. 調査対象のパブリックDNSからhost1.ghost.cs.dm.u-tokai.ac.jpのAレコードを検索する.
2. csゾーンからghostドメインを削除,namedを再起動する.
3. 別のパブリックDNSからhost1.ghost.cs.dm.u-tokai.ac.jpにアクセスできないことを確認する.
4. 調査対象のパブリックDNSからhost1.ghost.cs.dm.u-tokai.ac.jpにアクセスできることを（キャッシュが残っていることを）確認する.
5. ghostゾーンのネームサーバをdrosselからdrossel1に変更,namedを再起動する.
6. 600s以内に調査対象のパブリックDNSからdrossel1.ghost.cs.dm.u-tokai.ac.jp NSレコード検索をする.この応答のNSレコードは2.1節で述べた通り,1で受け取った応答と同じ信頼度を持つ.これによりキャッシュを上書きさせ,TTL値が延長する.
7. 600s過ぎた後に調査対象のパブリックDNSからhost1.ghost.cs.dm.u-tokai.ac.jpにアクセスできることを確認する.
8. 5～7を繰り返す

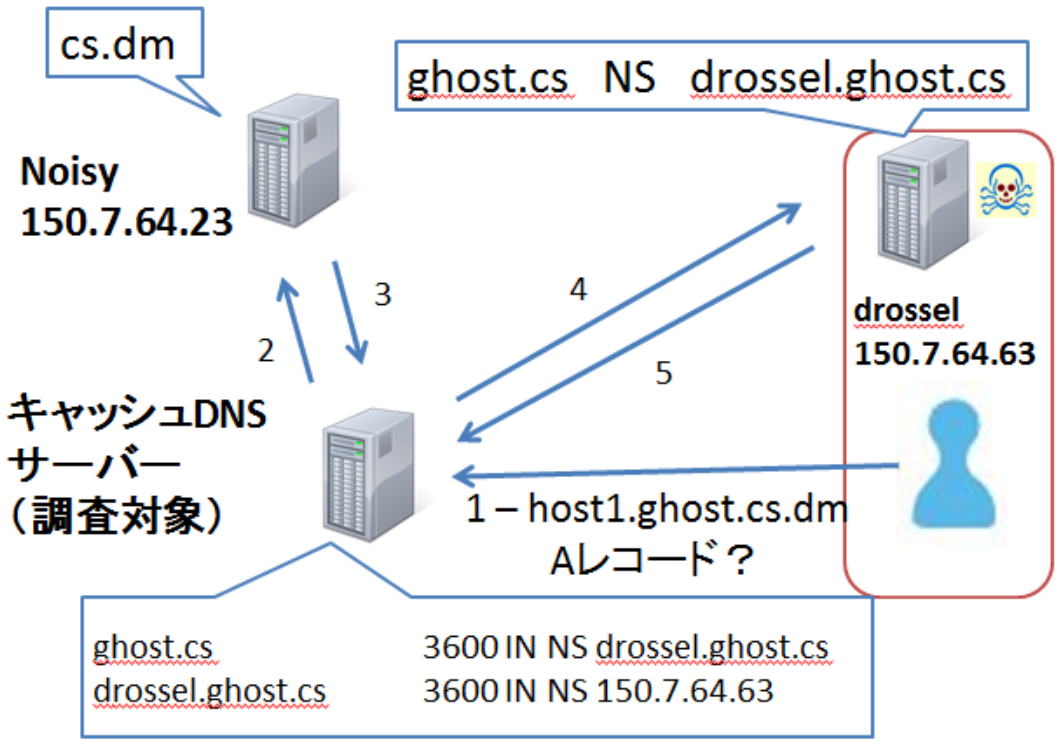


図 4.3 実験 1 の手順 1

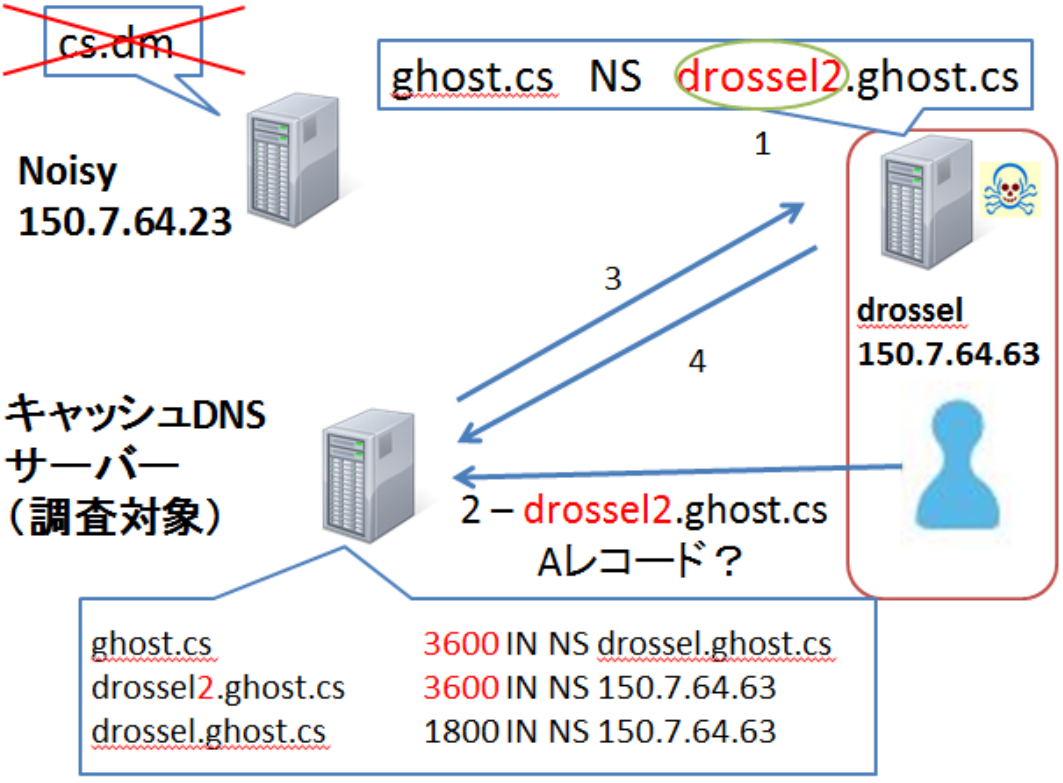


図 4.4 実験 1 の手順 2

手順 8 の繰り返しにはドメインの生存を確認する dlive.sh とネームサーバ名を変更し, named を再起動する refresh.sh を cron に登録し, TTL 値 3600s に合わせて 1 時間ごとに動作させている. それぞれを図 4.5, 4.6 に示す.

```
#!/bin/sh
cd /var/named/chroot/var/named
count=0
while read line; do
        array[$count]=$line
        count=`expr $count+1`
done < iplist.txt

i=0
para1=`date +"%m:%d:%k:%M"`
para2=`date +"%m%d%k%M"`
while (( $i < $count ))
do
        nslookup host1.ghost.cs.dm.u-tokai.ac.jp ${array[i]}
        i=`expr $i + 1`
        echo `date +"%m:%d:%k:%M"`
done > ghostlog/$para2.log
```

図 4.5 幽霊ドメイン生存確認スクリプト dlive.sh

```
#!/bin/sh
cd /var/named/chroot/var/named
para1=21`date +"%m%d%k%M"`

count=0
while read line; do
        array[$count]=$line
        count=`expr $count+1`
done < ghost.zone
array[2]="¥t¥t¥t¥t¥t$para1"
array[1]="@¥tIN SOA¥t$para1.ghost.cs.dm.u-tokai.ac.jp. root.ghost.cs.dm.u-tokai.
ac.jp. ("
array[7]="¥tIN¥tNS¥t$para1.ghost.cs.dm.u-tokai.ac.jp."
array[9]="$para1.ghost.cs.dm.u-tokai.ac.jp.¥tIN¥tA¥t150.7.64.63"
for i in "${array[@]}"
do
        echo -e $i
done > ghost.zone
/etc/init.d/named restart

count=0
while read line; do
        array[$count]=$line
        count=`expr $count+1`
done < iplist.txt

i=0
while (( $i < $count ))
do
        nslookup $para1.ghost.cs.dm.u-tokai.ac.jp ${array[i]}
        i=`expr $i + 1`
done
```

図 4.6 ゾーンファイル書き換えスクリプト refresh.sh

4.1.5　結果

実験結果を図 4.5 に示す.幽霊ドメイン脆弱性を確認できたパブリック DNS サーバは 2791 件中 2045 件であった.2 時間後には 869 件まで減少した.2013 年 1 月現在においても約 3 割の幽霊ドメイン名脆弱性を確認した.

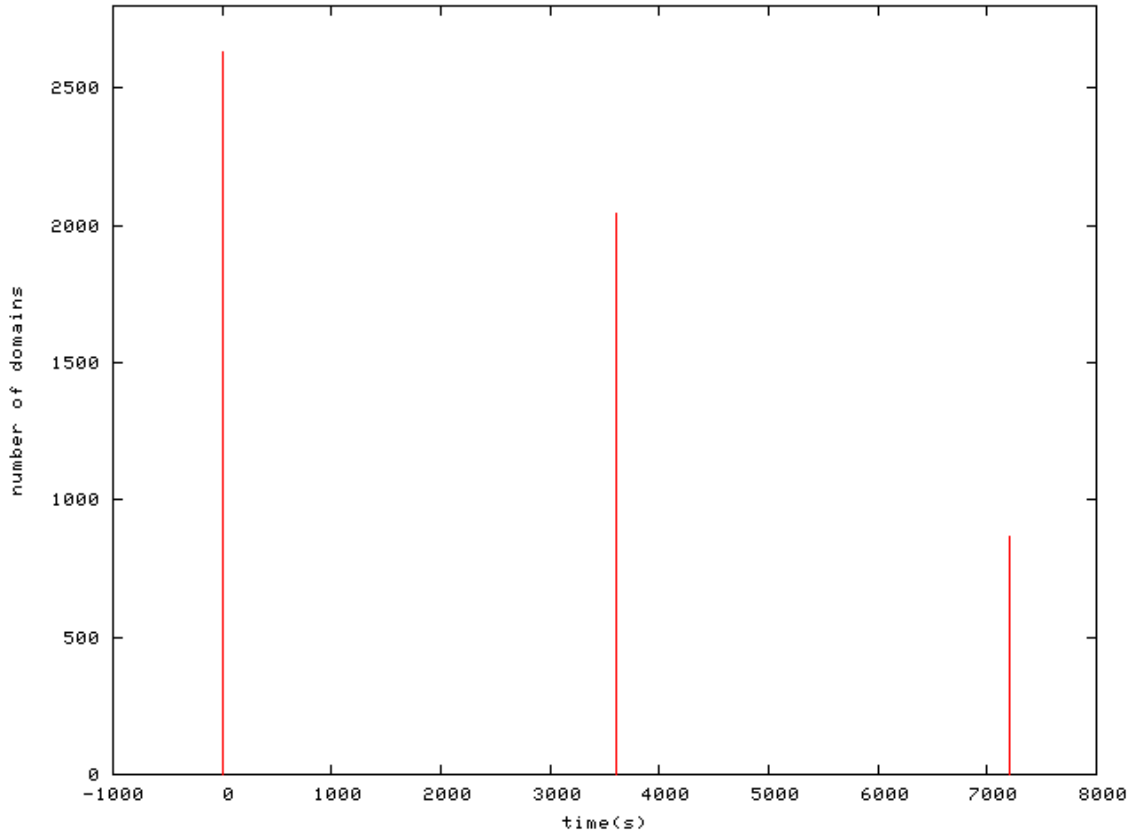


図 4.5　幽霊ドメインの生存数

4.2　実験２

4.2.1　概要

2013 年 1 月現在の主要パブリック DNS サーバの幽霊ドメイン名脆弱性への対応状況を明らかにするため,実験 1 により得たデータを基に,主要パブリック DNS の脆弱性を調査する.

4.2.2　結果

主要パブリック DNS の脆弱性の有無を表 4.3 に示す.Google 以外の主要パブリック DNS においては,2013 年 1 月現在においても脆弱性を確認した.GTEIDNS では 2 つの DNS のうちの 1 つからのみ確認した.

表 4.3 主要パブリック DNS の脆弱性

	IP アドレス	脆弱性
Google	8.8.8.8	なし
	8.8.4.4	なし
DNS Advantage	156.154.70.1	有
	156.154.71.1	有
OpenDNS	208.67.222.222	有
	208.67.220.220	有
Norton	198.153.192.1	有
	198.153.194.1	有
GTEI DNS	4.2.2.1	有
	4.2.2.2	なし

4.3 実験３

4.3.1 概要

2013 年 1 月現在の幽霊ドメイン名脆弱性を持つパブリック DNS サーバ割合が高い国を明らかにするため,実験 1 により得たデータを基に,国毎の幽霊ドメイン脆弱性を持つパブリック DNS を保有する割合を調査する.

4.3.2 結果

脆弱性を持つパブリック DNS の国毎の割合を表 4.4 に示す.ドメインの地理情報は public-dns.tk の情報に基づいている.イタリアが幽霊ドメイン名脆弱性を持つパブリック DNS サーバを最も高い割合で持つ国であった.割合の上位にはヨーロッパの国が多く見られた.

表 4.4　脆弱性を持つパブリック DNS の国毎の割合

	DNS 数	脆弱性有	割合
イタリア	74	27	0.365
日本	97	28	0.289
ベルギー	35	9	0.257
ドイツ	285	72	0.253
カナダ	38	9	0.237
中国	67	15	0.224
スウェーデン	509	108	0.212
アメリカ	888	180	0.203
オランダ	72	13	0.181
フランス	102	18	0.176
イギリス	212	33	0.156

4.4 実験４

4.4.1 概要

2013 年 1 月現在の主要 DNS ソフトウェアの幽霊ドメイン名脆弱性への対応状況を明らかにするため,実験 1 により得たデータを基に,主要 DNS ソフトウェア毎の幽霊ドメイン脆弱性保有率を調査した.

4.4.2 結果

実験結果を表 4.5 に示す.bind9.5 以前のものが最も高い割合で脆弱性を持っていた.bind9.6 以降にも約 6 割が脆弱性を持っていた.

表 4.5 主要 DNS ソフトウェア別攻撃成功率

	総 DNS 数	脆弱性有	割合
bind 9.5 以前	1169	1024	0.876
bind 9.6 以降	412	268	0.650
Power DNS	45	32	0.711
unbound	8	4	0.500
dnsmasq	12	9	0.750

4.5　実験５

4.5.1　概要

東海大学内 DNS の幽霊ドメイン名脆弱性への対応状況と幽霊ドメインの生存期間をあきらかにするため,それぞれ調査した.TTL は 3600 とした.学内 DNS の詳細について表 4.6 に示す.

表 4.6　東海大学内 DNS 詳細

IP アドレス	ドメイン名
150.7.3.5	ml.cc.u-tokai.ac.jp.
150.7.1.11	kamome.u-tokai.ac.jp

4.5.2　結果

実験結果を図 4.6 に示す. 2 つの DNS のどちらも 1 時間後の TTL 値の更新を許したため脆弱性を持っているが,2 時間後には更新されなかった.

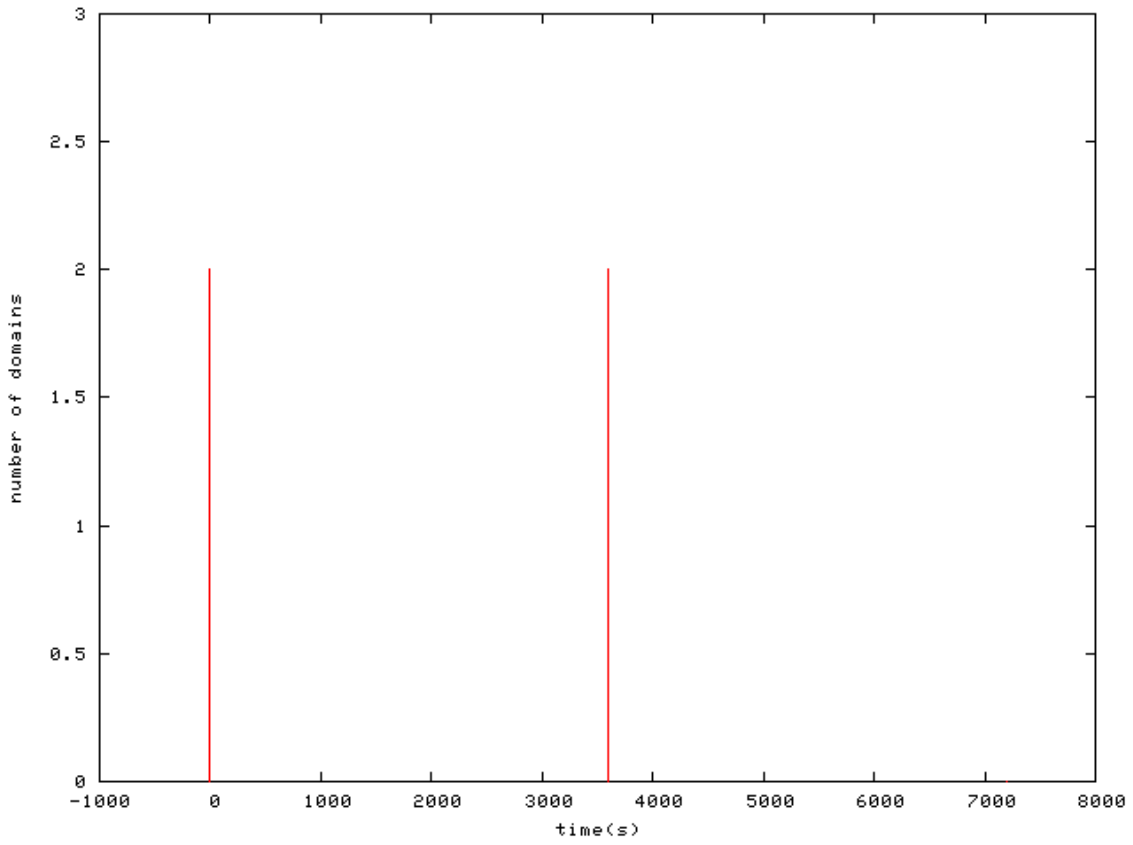


図 4.6 東海大学内 DNS における幽霊ドメイン生存期間

第５章　　考察

今回の脆弱性が発見されてから 11 か月程度経過した現在でも脆弱性が残っており,パブリック DNS はその危険性を理解した上で利用することがユーザーに求められる.主要パブリック DNS において脆弱性が確認されたが,幽霊ドメインが生存したのは 1 時間のみであった.Bind は幽霊ドメイン脆弱性に対してパッチを当てており,アップデートすることで脆弱性に対策することができる.Bind のパッチはキャッシュの更新を行う際に NS 名の更新は行うが TTL の延長は行わず値を保持させるものである.今回の実験において Google 以外の主要パブリック DNS は最初のキャッシュ更新を許しているため,Bind を用いていないか,パッチを当てずに異なる方法で対策を行っていると考えられる.幽霊ドメインの生存数が時間経過とともに減少していくのはキャッシュの定期的な削除を行っているためであり,キャッシュを削除する頻度が DNS 毎に分散しているため,段階的に減少していくと考えられる.幽霊ドメイン脆弱性を持つパブリック DNS の割合が一番多い国はイタリアとなった.割合が上位の国はヨーロッパ圏に多く見られた.

第６章　　結論

6.1　結論

DNS キャッシュ保持に関する脆弱性をついた攻撃に対するパブリック DNS の対応状況を調査した.パブリック DNS の約３割に脆弱性が残ることを明らかにし,その危険性を示した.主要パブリック DNS サーバにおいても脆弱性自体は残っていることを明らかにした.イタリアが幽霊ドメイン名脆弱性を持つパブリック DNS サーバ保有率が最も多い国ということを明らかにした.主要 DNS ソフトウェアにおいても 5 割以上が幽霊ドメイン名脆弱性を持つことを明らかにした.

6.2　今後の課題

本研究では TTL の値を 3600 に固定した実験を行ったが,TTL の値を変えて実験することでパブリック DNS キャッシュサーバの結果に変化がないか調べること,短い期間で行った実験を長期間行うこと,また実験にて明らかになった結果を用いて幽霊ドメイン脆弱性をついた攻撃を検知することを今後の課題とする.

参考文献

[1] Haixin Duan, Jian Jiang, Jinjin Liang," Ghost Domain Names: Revoked Yet Still Resolvable", NDSS Symposium 2012.

[2] RFC 2181, Clarifications to the DNS Specification, Internet Standard,1997.

[3] Public DNS Server List
(http://public-dns.tk/,).

[4] Yasuo Musashi, Kazuya Takemori, Shinichiro Kubota, and Kenichi Sugitani,"Detection of DNS Cache Poisoning Attack in DNS Standard Resolution Traffic", CSEC-53 2011.

[5]Kaminsky, "It's The End of The Cache As We Know it," 2008, http://kurser.lobner.dk/dDist/DMK_BO2K8.pdf.

[6] Ludeña Romaña, D. A., Kubota, S., Sugitani, K. and Musashi, Y.: DNS-based Spam Bots Detection
in a University, International Journal of Intelligent Engineering and Systems, Vol. 2, No. 3, pp. 11-18(2009).

[7] 堀 正義,中野 学,松本 勉,"複数の DNS キャッシュサーバを利用した安全な名前解決手法" ,ISEC2004-140.

[8] 「ghost domain names（幽霊ドメイン名）」脆弱性について
(http://jprs.jp/tech/notice/2012-02-17-ghost-domain-names.html,).

[9] BIND | Internet Systems Consortium
(http://www.isc.org/software/bind,).

[10] nlnetlabs.nl :: Bugs tracking ::
(http://www.nlnetlabs.nl/labs/bugs/,).

[11]PowerDNS - PowerDNS
(http://www.powerdns.com/content/home-powerdns.html,).

[12]Dnsmasq – a DNS forwarder for NAT firewalls.
(http://www.thekelleys.org.uk/dnsmasq/doc.html,).

謝辞

本論文を執筆するに当たり多くの方々から御指導,御鞭撻を賜りました.

特に,研究に関わらず私を導いてくださった東海大学情報通信学科通信ネットワーク工学科 菊池 浩明 教授に深く感謝を申し上げます.

また,本研究を遂行するにあたって御教示を賜りました東海大学情報理工学部情報科学科 内田 理 准教授に多大なる感謝を申し上げます.

そして,ともに励ましあい,議論を交わし,研究を進めてきた研究室の皆様に感謝を申し上げます.

最後に,ここまで支えてくださった両親に心から感謝するとともに,謝辞とさせていただきます.