DNS 보안 취약점과 DNSSEC & KeyperPlus

DNS(Domain Name System) 란

DNS는 인터넷의 기본이 되는 프로토콜(통신 규약)이며, 우리가 매일 수없이 사용하고 있습니다. 인터넷 통신을 할 때 IP 주소를 사용하는데, 이 주소는 기계가 인식하는 주소이므로, 사용자가 이 주소를 기억하는 것은 매우 불편 하므로, 사용자들은 IP 주소 대신 도메인 이름(Domain Name)을 사용합니다. 도메인 이름을 IP 주소로 변경하는 시스템이 DNS입니다. 도메인 이름을 IP 주소로 변경하는 과정을 DNS Resolution이라고 하기에, 이러한 기능을 하는 서버를 DNS Resolver라고 부릅니다

DNS를 이루는 Framework는 아래와 같은 3가지 부분으로 구성 되어 있습니다. .

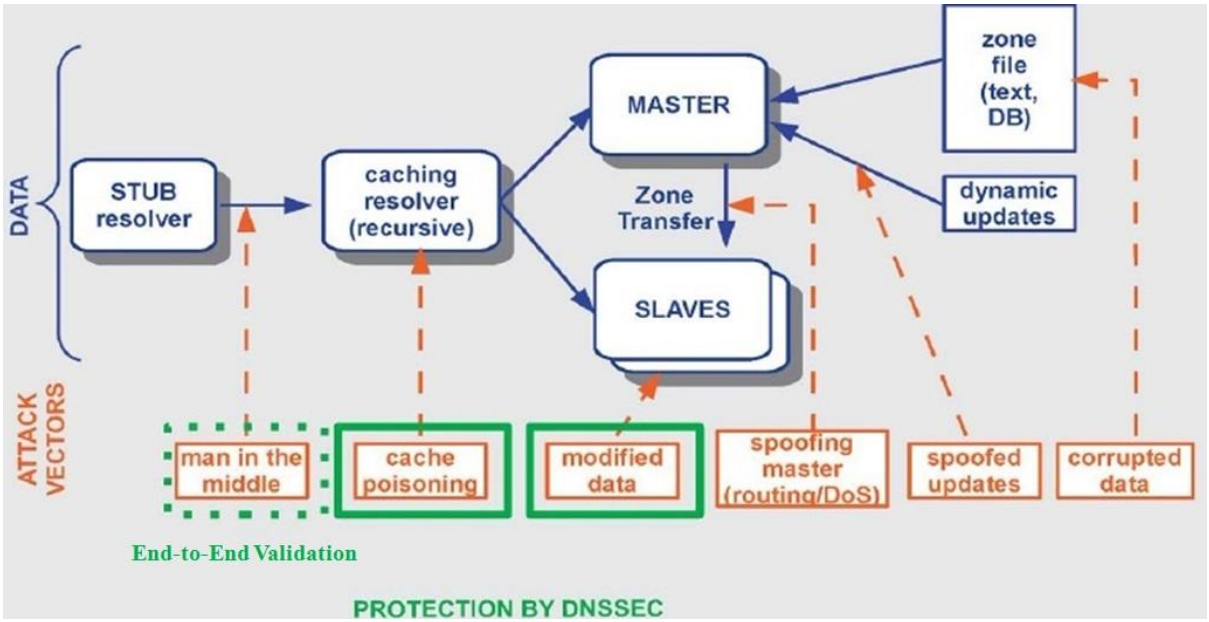
사용자 디바이스(PC, 스마트폰 등) <----> DNS 서버(DNS Resolver) <-------> 도메인 서버 (Authoritative DNS Server)

- 사용자 디바이스(PC, 스마트폰 등) : Stub Resolver라는 S/W가 설치되어 동작하며, 도메인 이름을 IP 주소로 변경하는 요청을 받아, 자신의 디바이스에 설정된 "DNS 서버" 주소로 보냅니다.
- DNS 서버(DNS Resolver) : DNS query를 받아, 도메인 이름에 대한 IP 주소를 알아 냅니다. 이 때, Root, TLD(Top-Level-Domain), 해당 도메인을 순차적으로 찾아가기에, Recursive Name Server라고 부르기도 합니다. 한번 찾은 IP 주소를 메모리에 일정기간 동안 저장한다고 해서 Caching Resolver라고 부르기도 합니다.
- 도메인 서버(Authorative Name Server : 자신의 도메인에 대한 IP 주소를 가지고 있습니다. 3-level 이상으로 이루어져 있습니다(Root, TLD, Lower-level-domains). 각 도메인 서버는 구현상 1개의 Master 서버와 다수개의 Slave 서버로 되어 있습니다.

DNS 보안 취약점

DNS 개발 당시, 보안은 충분히 고려하지 않았고 효율적인 측면만을 강조하여 만들었기 때문에, 태생적으로 보안 취약점을 가지고 탄생되었습니다.

아래 그림은 현 DNS 시스템의 보안 취약점 즉 Attack Vector(해커의 공격 지점)를 보여 줍니다.



.

DNS Cache Poisoning 공격이란.

보안 취약점을 이용한 대표적인 공격이 DNS Cache Poisoning 공격입니다.
도메인 이름에 대한 IP 주소를 알아내어 응답하는 것을 좀 더 효율적으로 처리하기 위해, 한번 변환된 주소 테이블을 자신의 메모리에 저장해 두고, 동일한 주소변환 요청이 오면, 전체 프로토콜을 거치지 않고 테이블에 저장된 값으로 답변을 합니다. 이 때 해커가 이 테이블에 접근하여, 특정 도메인 이름과 대응되는 IP 주소를 자신이 의도하는 가짜 IP 주소로 조작하면, 사용자는 해커가 유도하는 가짜 사이트에 접속하게 됩니다. 즉 DNS 서버의 메모리에 저장된 IP 주소를 조작하는 것을 DNS Cache Poisoning 이라고 합니다.
그러면 해커가 어떻게 데이터를 조작하는 가 인데, 여기에 핵심이 있습니다. 즉 DNS 서버에 접근하기 위한 특별한 침투 기술이 있는 것이 아니라, DNS 서버에 일련의 조작된 Data를 DNS 프로토콜 양식에 따라 송신만 하면 자동적으로 이루어 집니다. DNS 서버는 프로토콜에 따라 동작하기 때문에, 자신이 받는 Data가, 해커가 보내는 조작된 data 인지 아니면 적법한 도메인 서버에서 주는 Data인지 분간하는 기능이 없기 때문입니다.
DNS 서버 주소는 거의 대부분 공개가 되어 있기 때문에, 악의적 의도를 가진 해커는 특정 DNS 서버에 저장된 Cache Data를 조작하는 공격을 할 수 있습니다..

기술적으로 개념만 설명하면, DNS 서버는, DNS 프로토콜에 따라 주고 받는 데이터 양식이 있는데, DNS transaction ID(16 bit) 와 Source Port Number 를 가지고 확인을 합니다. 해커가 DNS 서버에 DNS transaction ID 와 Source Port Number가 동일한 조작된 Data를 송신하면 됩니다. DNS Transaction ID 와 Source Port Number를 동일하게 예측하는 것은 쉽지 않은 일입니다만, 컴퓨터를 통하여 무작위로 만들면 어느 순간에는 일치가 되니까요...즉 시간의 문제입니다. 요즈음은 컴퓨터 성능이 좋아져서 일치가 되는 시점이 더 빨라 질 것입니다.

DNS Cache Poisoning 공격의 목적

Fake(가짜) Site에 redirect시켜, 해커는 아래와 같은 목적을 수행 할 수 있습니다.

1) 파밍 (Pharming) : 금융정보(ID, Password 등) 탈취 목적
2) 고객 PC를 Control할 목적으로 악성 S/W 유포 목적 : 고객은 자신이 신뢰하는 Site이므로, 보안프로그램으로 가장하면 아무런 의심 없이 download 받게 됨을 이용
3) 정치적 목적으로 특정 Web Page로 Redirect
    - 2013년 10월 10일 "www.google.com.my" site가 파키스탄에 있는 Fake website로 Redirect
    - 2013년 8월 New York Times Website가 시리아 전자 군대가 Control하는 Web Page로 Redirect
4) 이메일 하이재킹(Hijacking)
    - 해커가 지정한 이메일 서버로 이메일이 보내지게 되며, 그 후 원 수신자에게 보내지게 되므로 메일의 송신자 및 수신자는 이메일이 감청당한 일 또는 변조 당한 일을 알 수 가 없습니다.
    - 2014년 카네기 멜론 대학의 CERT/CC 팀에서 이러한 가능성을 발표하였습니다.
5) VoIP 통신 도청 및 감청
    - 오스트레일리아에서는 VoIP 사용시, 이러한 문제점이 있다는 것을 인지 하였기 때문에 VoIP 장비의 필수 기능으로 DNSSEC 기능을 규정 하였다고 합니다.

DNS Cache Poisoning 공격이 가능한 DNS 취약점 사례

사례 1) 1997년 7월 Eugene Kaspureff가 발견하였습니다.

DNS 서버가 DNS query에 대한 응답의 additional section에 있는 data도 Cache entry에 추가하는 기능(취약점)을 이용하면 공격이 가능하였습니다(공격자가 자신이 만든 도메인 서버에 대한 DNS query를 DNS 서버에 보내고, 공격자가 만든 DNS 서버는 응답할 때, additional section에 타 site 주소에 대한 변조된 data를 추가하여 응답하는 방식). 이에 대한 대응으로 "bailiwick(범위)"에 속하지 않는 data는 Cache entry에 추가하지 않도록 패치 하였습니다.

사례 2) 2007년 6월 Amit Klein이 발견하였습니다.

DNS message ID field의 길이가 16 bits 라는 기능(취약점)과 도메인 서버에 따라 random 값이 truly random하지 않고 예측 가능한 로직으로 되어 있어서, 다음 번 Message ID를 계산으로 예측할 수 있다고 합니다. 따라서 예측한 응답 Message를 다량으로 보내면, 그 중의 하나는 일치할 수 있으므로, 이 기능을 이용하면 공격이 가능하였습니다. 이에 대한 대응으로 Message ID를 예측하기 어렵게 더 random화 시키는 방법으로 패치 하였습니다.

사례 3) 2008년 8월 Dan Kaminsky가 발견 하였습니다

Message ID(Query ID: QID)를 truly random 하게 하더라도, 매우 정교한 공격으로 QID가 일치하는 케이스를 만들 수 있다고 합니다. 이 기능을 이용하면 공격이 가능하였습니다. 이에 대한 대응으로 UDP Source port randomness를 추가 했으며, randomization bits를 추가하는 패치를 하였다고 합니다.

사례 4) 아직 보고 되지 않고 있습니다만,

UDP Source port randomization을 추가한 것은 공격자가 들어오지 못하도록 장벽을 높이 세운 것에 지나지 않는다고 하며, 또한 Firewall 뒤에 있는 DNS Resolver는 UDP Source port randomization의 효과를 누리지 못한다고 합니다. 그리고 성능 좋은 컴퓨터로 무작위 UDP Source Port를 Generate 시켜, 일치시키는 Response를 만들어 낼 수 있다고 합니다. 따라서 DNS Cache Poisoning 취약점은 현 DNS 프로토콜로는 해결할 수 없는 근본적인 문제라고 합니다.

DNS Cache Poisoning 공격 피해 사례

1) 2009년 4월, 브라질 대형 은행 중의 하나가, DNS Cache Poisoning 공격의 희생양이 되었다고 합니다. 특정 브라질 ISP의 DNS 서버가 DNS Cache Poisoning 공격을 당하여, 이 은행에 접속하고자 하는 사용자를 가짜 사이트로 접속하게 하였다고 합니다.
2) 2010년 12월 러시아의 가장 큰 온라인 Payment 사이트(ChronoPay)가 가짜 사이트로 Redirect 되었다고 합니다.
3) 2011년 11월 브라질은 Massive DNS Cache Poisoning 공격을 받았다고 합니다.

위와 같은 사건으로 인해, 브라질은 Financial industry에 속한 모든 Entry-Level Domain에 DNSSEC을 지원하는 것을 완료 했다고 합니다.

DNSSEC : DNS Cache Poisoning 공격에 대한 대응책

기술적으로 DNS Cache Poisoning 공격에 대응하는 Solution으로 나온 것이 DNSSEC(DNS Security Extension) 입니다. 즉 DNS 보안 취약점을 해결하기 위해서는 DNSSEC을 구현하여야 한다고 권고하고 있습니다. 현재로선 DNS Cache Poisoning을 근본적으로 해결하는 Solution은, DNSSEC 구현 외에는 다른 대안이 없다고 합니다.

DNSSEC란, DNS 서버로 하여금 도메인 서버로부터 받은 Data를 확인 하도록 하는 것입니다. 이것을 DNS 서버의 Validation 기능 이라고 합니다. 확인 할 때 PKI 암호 기술을 사용하여 데이터 위조 및 변조를 발견해 냅니다.

사용자가 DNS 서버에게 DNSSEC Validation을 요청한 경우, 해당 정보가 변조되었을 경우, DNS 서버는 Validation에 실패하게 되고, IP 주소를 알려주지 않습니다. 따라서 사용자는 가짜 Site로 접속을 하지 않게 됩니다.

2005년도에 DNSSEC Final version이 규정되었지만, 실제로 DNSSEC가 제대로 구현되기 시작한 시기는 2010년이라고 보시면 됩니다. Root Domain을 관리하는 ICANN 에서 2010년 6월에 Root Domain에 DNSSEC구현을 완료 했기 때문입니다..

DNSSEC의 구현

DNSSEC 표준을 만든 후, 이를 인터넷 산업계에 구현하고자 많이 노력하였으나, 여러가지 이유로 구현되어지는 속도가 매우 느린 상태에 있습니다. PKI 기술을 사용하여 Trust-Chain 방식으로 구현되어 지기 때문에, Root 기관과 중간 단계 기관, 하위 기관 모두 구현이 되어져야 합니다. Root 기

관인 ICANN 과 Top-Level-Domain(.com , .net , .kr , .jp 등) 기관들은 이미 구현이 되어져 있습니다. 하지만 하위 기관(예를 들면, naver.com , kn.co.kr)에 속하는 Domain 소유자 중 DNSSEC을 구현하고 있는 비율은 매우 적습니다. 예를 들면 .com 도메인인 경우, 전체 도메인 수는 1억개가 넘는 다고 합니다. 이 중에서 DNSSEC를 구현한 곳은 이제 50만이 조금 넘습니다(.com을 운영하는 Verisign 사에서 DNSSEC Scoreboard Site[scoreboard.verisignlabs.com]를 통하여 실시간으로 보여주고 있습니다). 아마도 대부분 공격을 당해도 피해를 볼 것이 없기에 투자할 필요성을 느끼지 못하는 것 같습니다. 하지만 피해를 방지하기 위하여 DNSSEC관련 투자를 한 Site 수도 50만을 넘었습니다.

DNSSEC 시스템이 제대로 동작되기 위해 구축 되어져야 할 3가지 분야

DNSSEC이 동작되기 위해서는 DNS Infrastructure 의 3가지 파트에서 모두 DNSSEC을 지원해야 합니다.

사용자 디바이스(PC, 스마트폰 등) <----> DNS 서버(DNS Resolver) <-------> 도메인 서버 (Authoritative DNS Server)

- 사용자 디바이스 : DNS 요청을 할 때 DNSSEC validation을 요청해야 함 (Windows 7 부터 지원, Default로 Off 이나, 제공되는 설정 S/W[gpedit.msc] 를 사용하여 원하는 도메인만 ON 시킬 수 있음)
- DNS 서버(DNS Resolver) : DNSSEC data를 validation하는 기능을 지원해야 함
- 도메인 서버 : 도메인 zone data를 암호키로 signing 해 두어야 함, 업체의 규모나 기능에 따라, 암호키 보관을 위해 HSM 장비 사용을 고려해야 합니다(ICANN과 TLD는 HSM 장비를 사용하고 있습니다)

도메인 서버(Authorative Name Server) 의 종류

- Root : ICANN 에서 운영
- TLD(Top Level Domain) : ".com" , ".net" , ".org"와 같은 gTLD(generic TLD) 와 ".kr", ".jp" 와 같은 ccTLD(country code TLD) 가 있으며, ".com" 과 '.net" 는 Verisign 이라는 회사에서 운영하며, ccTLD 는 각 나라의 국가기관에서 운영
- Lower Level Domain : 각 회사나 기관에서 운영 (".naver.com" , ".kn.co.kr" , 등등)

도메인 서버를 구현하는 방식

- Linux(UNIX)/Windows 서버에서 BIND S/W 와 같은 DNSSEC 지원 S/W를 설치하여 구성
- DNS 전용 Appliance 사용

DNSSEC 구현에 사용되는 암호키 보관 방식

- S/W 방식 (보안에 약함)
- HSM 장비를 사용
  - DNS 전용 Appliance 사용시에도 보안을 더 강화하기 위해서는 추가로 HSM 장비를 부착하여 사용합니다.

HSM 장비의 종류.

- PCI-HSM 카드 (속도는 빠르나 상대적으로 보안에 약함)
- TCP/IP로 통신하는 전용 HSM 장비

KeyperPlus

저희 회사(AEP-코리아네트)는 영국 AEP(www.ultra-aep.com)사의 HSM 장비인 KeyperPlus를 국내에 공급하고 있습니다. KeyperPlus는 시장에 나온 제품중 보안 레벨이 가장 뛰어난 FIPS 140-2 Level 4 인증을 받은 제품입니다.

AEP사의 Keyper/KeyperPlus를 사용하고 있는 기관 및 회사로는 Root Domain을 운영하는 ICANN과 ".com" 과 ".net" 를 운영하는 Verisign 를 비롯하여 US Military, Raytheon 등이 있으며,많은 나라의 ccTLD 에서 Keyper/KeyperPlus를 사용하고 있습니다.

ICANN이 Keyper를 선택한 이유는 시장에 출시된 제품 중 보안 Level이 가장 뛰어나기 제품이 때문이라고 하였습니다.

참고사항으로, 현실적으로 고민하고 있는 2가지 분야도 언급하고자 합니다.

DNSSEC Validation을 해야 하는 위치에 대한 고민

DNS Cache Poisoning 공격만을 막기 위해서는 DNS 서버(DNS Resolver)에서 DNSSEC Validation을 하면 충분 하지만, 사용자 디바이스와 DNS 서버 사이에서 중간자 공격(Man-In-the-Middle Attack)을 하면, 즉 DNS 서버에서 사용자 디바이스로 가는 Data를 중간에 가로채서 변조하면, 사용자 입장에서는 해킹을 당하는 것이므로, 이러한 공격 조차 막으려면, 사용자 디바이스에서 DNSSEC Validation을 해야 합니다. 사용자 디바이스에서 DNSSEC Validation을 지원 하는 S/W Solution이 아직 보편화되지 않은 것이 현재 상태입니다.

사내 DNS 시스템에서도 DNSSEC을 구현해야 하는 가 ?

사내 DNS 시스템인 경우는 외부인이 접속하지 않고 사내에서만 사용하기 때문에, 외부로 부터의 해커 공격은 없지만, 내부 공모자에 의한 공격은 있을 수 있으므로, 민감한 정보를 다루는 기관 및 기업들은 내부 공모자에 의한 공격을 방지하기 위하여, DNSSEC을 구현할 필요가 있다고 합니다.

AEP KoreaNet

서울특별시 금천구 가산디지털1로 168 우림라이온스밸리 C동 301호 www.kn.co.kr