Internet 붕괴의 기술적 원인 분석

==========

Internet 마비에 대한 기술적인 원인 분석
(공격기법 분석 갱신 Ver. 1.0)

2003년 1월 26일

(revision 0.2 2003년.1월 27일)

(revision 1.0 2003년.1월 28일)

작성자: 전 상훈 (nickname: 바다란)
(winsnort@securityindepth.net)

Internet 붕괴의 기술적 원인 분석

==========

@ 개요

revision 1.0 개요:
여러 곳의 조언과 수많은 질문과 정보를 총 정리 하여 새로운 가설을 세워 보았다. 이 가설이라면 모든 문제에 대한 설명이 일목요연해 진다. 우선 공격 방법에 대한 근본가정 자체를 달리 해 보았다. 또 피해 현상에 대한 상황들을 종합 하였고 이 것을 바탕으로 하여 새롭게 공격방법과 현상에서 드러난 모든 결과를 총괄해 보도록 하겠다. 어떻게 인터넷 인프라가 붕괴가 되었으며 KT DNS 다운현상이 일어 났는지 설명이 가능한 논리이다. 개인이 작성하는 것이므로 많은 이견이 있을 수 있다. 또 이전 발표 문서에서의 DNS 서버의 붕괴를 설명하는 관점도 완전히 다른 각도에서 보게 될 것이고 공격에 사용된 기법과 내가 만든 가설의 문제점도 조명이 될 것이다. 더불어 Keep alive packet의 전송에 따른 SQL 서버의 Resource 소모에 관련된 내용도 폐기 처분한다. 여러 능력 있으신 분들의 로그결과와 dump 결과 그리고 신뢰할 만한 몇몇 곳에서의 점검결과를 확인 하여준 결과물에 keep alive 패킷의 전송내역이 없었으므로 이 가정은 무용지물이 되어 버렸다. (실제로 트래픽을 보거나 여러 다양한 경로에서의 과정을 알 수 있는 위치 였다면 보다 빨리 현재 쓰고 있는 가설이 나왔을 것을 의심하지 않는다.)

자 그럼 위의 모든 가정을 종합했을 때 결과는 어떻게 나오며 개인적인 상상력으로 판단해 보건대 어떤 의도 였을까?

공격자는 의도적으로 대한민국 인터넷 인프라의 붕괴를 노렸다고 판단한다. 이번 인터넷 붕괴를 노린 공격자는 우리나라가 자랑할 만한 몇 안 되는 것 중의 하나인 인터넷 인프라를 붕괴함으로써 혼돈을 유도하려 했던 것이 목적으로 보이며 누가 했느냐 하는 것은 이러한 목적을 통해 이익을 취하거나 큰 규모의 시기심이 있는 덩치 큰 이웃사촌(남 잘되는 것 배아파 하는)이나 욕심쟁이들이 대상이 될 것이다. 우리나라도 드디어 남들이 부러워할 만한 그런 위치에 현재 서 있는다는 반증이다. 그러면 어떻게 했을까?..이건 가설을 보며 이해 하도록 하자. 또 이 가설을 뒷받침할 만한 공격도구 들도 이미 파악했다. 실제 구축 환경이 가능하신 분은 직접 테스트 해보시기 바란다.

결과론 적으로 KT DNS 다운 및 인프라의 붕괴에 영향을 미친 것은 세 가지 정도 이다.

1. DNS 서버에 대한 Ddos Attack. (Ping flooding이 아닌 DNS query에 의한 공격)
2. SQL Overflow 웜에 의한 데이터 증가
3. 2천만에 이르는 국내의 네티즌과 컴퓨터

Internet 붕괴의 기술적 원인 분석

==========

자 그럼 왜 이런 영향을 미치는 요인과 목적, 인프라의 붕괴가 오게 되었는지 들어가 보도록 하자.

0.2 version 문서는 : http://www.securityindepth.net , http://www.securitymap.net
0.1 version 문서는 : http://my.netian.com/~mil21/internetcrisis0.pdf

==========

revision 0.2 개요: IDS 로그 및 기타 로그를 확인해 보았을 때 1433번 포트로의 접속은 거의 없었으며 1434번 포트로의 UDP 접속이 거의 대부분이라 공격 패턴 판단을 잘못 하고 있다는 생각 하에 재갱신 하게 되었습니다. 공격패턴을 수정합니다. 시나리오도 수정 했습니다.

==========

Resolution service 개략:

SQL 서버에서 공격 대상이 된 서비스는 Resolution 서비스로서 이 서비스에 대해서 알아둘 필요가 있다. MS SQL 서버에서 Data를 저장하는 영역을 여러 개로 나눌 수 있는데 이 하나의 영역 자체를 instance라 칭하며 SQL 서버와의 통신에 사용하는 포트는 1433번 포트로만 사용이 가능하다. 하나의 SQL 서버에 여러 개의 Instance가 존재 한다는 말은 한대의 SQL 서버로 여러 개의 Database를 나누어서 사용한다는 말과 동일하다 할 수 있다. 그러면 여러 개의 Database가 하나의 서버에 존재할 때 통신을 할때 어떻게 통신을 할 수 있을까?. 1433번 포트에 Listening하는 데이터의 Instance는 단 하나이며 여러 개가 있을 경우에는 특정 포트에 접속하여 찾는 방식이 아닌 1434번 포트를 이용하여 다른 Instance를 찾게 된다. 이 1434번 포트가 Resolution 서비스가 된다.
만약 한대의 서버에 a 라는 쇼핑몰 a 사의 DB와 B라는 쇼핑몰 B사의 DB가 동시에 존재할때 먼저 설치된 a 사의 DB만 1433번에서 listening 하고 있는 것이 된다. 그럼 다른 DB는 어떻게 될까?
1433번 포트가 아닌 다른 임의의 포트에서 listening 하도록 세팅이 가능하도록 되어 있다. 하나 하나의 별개의 DB를 instance라고 하며 이 임의의 포트에 배정된 instance를 찾기 위해 Resolution 서비스가 이용이 된다. 이 instance는 로컬에 있는 것 뿐 아니라 원격의 instance도 동일하게 해당이 된다.

✍✍1.27일 추가정보 추가: 1434번 포트는 두 가지 용도로 사용이 되고 있음을 추가로 확인 했다. 먼저 들어오는 트래픽에서 첫번째 바이트가 0x04일 경우에 Monitoring thread가 실행이 되며 Keep –alive 매커니즘의 경우 단일 바이트로 0x0A 시그널을

Internet 붕괴의 기술적 원인 분석

==========

받았을 경우에 동일한 0x0A 의 single byte packet을 반응함으로써 매커니즘이 구현되어 있다.

자 설명이 길었다. 이정도면 Resolution 서비스에 대한 설명은 충분히 되었으리라 본다. 그러나 이 Resolution 서비스는 단지 잼머용 데이터와 트래픽 폭주를 일으키기 위한 용도로만 사용이 되었으며 DNS 서버의 다운을 설명하지 못한다. 그 이유는 분명하게 짚어 줄 것이다. 또한 이 SQL 서버의 Resolution 서비스 자체가 DNS 다운의 이유에 해당하지 못함을 지적할 것이다.

1월 28일 오후 10시까지의 결론은 어제의 결론과 동일하다. 1434 UDP port 한 곳만으로 전파 및 무한 루프에 의한 리소스 소모가 일어난다고 판단한다. 1434 UDP port에 대해 들어오는 패킷에 따라 가변적으로 작동하는 SQL Server의 매커니즘으로 인하여 적지않은 혼동이 있었다. Packet의 첫번째 byte가 0x04로 시작할 경우에 NGSSoftware사의 분석 문서를 보게 되면

전송 문자열: \ 0x04 \ 0x41 \ 0x41 \ 0x41 \ 0x41 (0x41=A) 이런 패킷이 전송 되었을 때 첫번째 패킷이 0x04로 시작 되므로 Monitoring thread가 실행이 되며 이 thread는 레지스트리의 값을 읽는다고 한다.
HKLM \ Software \ Microsoft \ Microsoft SQL Server \ AAAA \ MSSQLSERVER \
CurrentVersion (등록된 SQL Server의 instance를 찾는 과정이라 예상됨)

✍✍1.28일 추가 확인 사항 (무지 어렵게 찾았음)
전송문자열이 \ 0x04 \ 0x41 \ 0x41 \ 0x41 \ 0x41 (0x41=A) 에서 A라는 문자가 가지는 의미는 무엇일까? 또 버퍼오버런이 일어나는 서비스가 과연 정상적인 작동을 할 수 있을 것인가 하는 의문이 들었다.

여기에서 AAAA라는 문자열은 instance를 말한다. 즉 UDP 1434 포트에 0x04로 시작하는 패킷을 보내고 그 뒤에 긴 문자열의 instance를 쿼리 한 것이다. 즉 이 패킷을 받은 서버에 SQL Server instance로 AAAA 라는 인스턴스를 찾으라는 명령이 되겠다. 당연히 0x04 포트로 시작되면 Resolution 서비스가 활성화 되면서
HKLM \ Software \ Microsoft \ Microsoft SQL Server \ AAAA \ MSSQLSERVER \
CurrentVersion 과 같이 로컬 레지스트리를 검사하게 될 것이다.
(SQL 서버 사용하시는 분은 DB 여러 개를 등록 하셨을 경우 레지스트리 키 값을 확인 해보시기 바란다.)

여기에 AAAA 라는 인스턴스가 있을 경우에만 이 인스턴스가 존재한다고 반응을 하게 될

Internet 붕괴의 기술적 원인 분석

==========

것이다. 그런데 여기서 중요한 착오가 이전의 가설에서 발생한다. 인스턴스 문자열의 길이가 생각보다 길었고 Resolution 서비스는 인스턴스 문자열의 길이를 체크하지 못한 이유로 인하여 정상작동이 되지 않으며 버퍼오버런 되면서 return address를 반환하게 될 것이다. 웜은 이 return address를 이용하여 SQL 서버가 실행되는 권한으로 (대부분 System 권한 혹은 Domain User 권한) 패킷에 실려 있는 공격 코드를 실행하게 되는 것이다. Resolution 서비스는 정상적인 작동이 시작 되기도 전에 비정상종료가 되는 것이 정확하다.

다시 언급하면 Resolution 서비스는 정상 실행 될 수 없다. 만약 다른 대체적인 SQL Server의 기능을 통해 instance를 찾는 과정이 진행 된다 하여도 이 것은 몇몇 분들이 보내주고 알려주셨던 조언처럼 DNS의 구조에 역행하는 형태로 KT DNS를 침해 할 수 없음을 분명히 한다. 이러한 형태로 KT DNS 가 붕괴 되려면 그 이전에 모처에 안전하게 모셔져 있는 root server도 다운이 되었어야 했다. KT DNS의 장비가 떨어 질리는 더더욱 없기에 말이다. 국내 DNS 관련 트래픽의 54프로 가량을 처리하면서도 그동안 안정적이였던 서버가 다운이 될 경우에는 전 세계에 흩어져 있는 DNS root 서버도 당연히 막대한 타격을 받았어야 했다. 그러나 유감스럽게도 그런 소식은 뉴스와 신문에도 보안 메일링 리스트에도 전혀 없었다. 하물며 동네 DNS 조차도 죽지 않았다. 다만 트래픽이 올라 갔을뿐.
이 트래픽이 올라 간 것은 또 설명할 기회가 있을 것이다.

일전에 발생한 root server에 대한 대대적인 ping flooding 공격에 대한 기사는 여기서 확인 가능하다.
http://siliconvalley.internet.com/news/article.php/1486981 작년 10월 23일이다.

대부분의 사람들 및 보안업체가 웜 코드인 404byte만 패킷 모니터링 되고 다른 이상징후는 보이지 않는데도 불구하고 인터넷이 마비 된 것에 대해 404 byte의 위력이라고 생각을 하실 것이다. 또 다른 나라는 404byte의 1434번 포트로의 감지만 되면 DNS 서버 다운이나 인터넷 마비등의 사태는 없었다고 생각하면 우리나라의 현실에 대해 아주 절망적인 생각을 가지게 될 것이다. 그러나 본인은 그렇게 생각 안한다. KT 관계자들 고생 무지 많이 했을 것으로 짐작한다. 2시간 만에 DNS 서버 용량 3배로 늘리려면 엔지니어들과 테크니션들의 엄청난 노력 없이는 불가능한 일이다. 그리고 그것이 곧 뛰어난 능력이다. 장담 하건대 본인이 제시하는 시나리오 대로 다른 나라에도 비슷한 일이 일어난다면 국내 엔지니어들과 테크니션들의 대단한 능력을 알게 될 것으로 믿어 의심치 않는다.

우리가 간과하고 있는 것의 가장 큰 하나는 404byte에 포함된 비밀일 것이다. 사실 별거 없다. 이 웜이 대단한 것은 트래픽 증가에 아주 효율적이며 UDP 패킷이므로 하나하나의 패킷을 drop 시켜야 한다는 것이 방화벽에 부담을 준다는 것. 그 이외에 이 웜에 포함된 것은 없다. 웜의 분석에서도 나오지만 앞부분은 웜 | EIP|garbage 형태로 들어오는 것을

Internet 붕괴의 기술적 원인 분석

==========

Reverse로 전환을 하는 부분이고 그 다음은 1434번 포트로 웜의 전체를 (404byte) 보내기 위해 다시 버퍼에 싣는 과정이다. 마지막으로 TickCount에 따라 임의의 IP Address에 이 내용을 발송한다. 발송하는 과정을 계속 반복 하는 루틴을 돌며 IP 주소의 값을 변경함으로써 상당히 빠른 형태로 지속 실행이 되는 것 이것이 웜의 정체다.

업계에 있는 분들이나 왠만한 분들은 다 아시는 상황 일 텐데 네트웍에 모니터링 되는 것은 웜 밖에 없는데도 불구하고 DNS가 죽고 인터넷이 마비되었다는 사실에 조금 황당함을 느끼실 것이다. 본인도 본인이 생각할 수 있는 유일한 DNS 서버에 대한 공격 가능 로직이 SQL Resolution service의 instance 관련된 추정 외에는 없을 정도로 빈약함에 시달려야 했다. 또 수시로 들어오는 메일과 메신저로의 질문에 조차 DNS 다운에 대한 질문을 하시고 의문을 표시하시는 고수 분들을 만날 때 마다 너무나도 빈약한 본인의 논리에 상당한 자괴감을 느낄 정도 였다.

완전히 생각을 바꾸어야 한다고 생각한 계기는 Resolution 서비스 자체가 비정상종료 될 수 밖에 없다는 instance 쿼리 로직을 알고 난 뒤이다. 그리고 지속적으로 정보를 제공받고 고달프게 타인을 괴롭히면서 얻어낸 로그와 패킷의 캡쳐 등을 확인한 결과 대부분이 404byte의 1434번 UDP 트래픽 이였으며 그 외 잡다한 로그들은 평상시 보던 패턴과 유사함을 거듭 확인 하게 되었다. 또 모 ISP 관계자와의 장시간의 채팅을 통해 실 데이터를 보지 못하고 접할 수 없었던 정보를 듣고 난 후 본인은 다소 위험한 가설을 세우게 되었다. 만약 이 가설이라면 이 상황을 설명할 수 있을 것이라 판단하게 되었고 이 가설을 뒷받침할 만한 근거들을 찾아 헤맸다. 그리고 찾았다.
장황하게 늘어 놓은 사설은 지금 어느 정도 수습 단계에 또 다시 충격을 줄 수도 있다는 개인적인 생각으로 주절 댄 것이니 양해 바란다.

본인이 생각한 가설은 이렇다.

다수의 DNS query Ddos Attack -✍ KT DNS -✍ DNS 처리 용량 증가 --> KT ADSL 사용자 및 KT DNS 사용자 인터넷 접속 과정에서 느림을 인지 (domain name을 IP로 변환하는 과정이 느린거다 인터넷 라인은 무사했다.) ✍ DNS 사용자 들의 refresh 과다 반복 (DNS Query의 급격한 증가) ✍ DNS 서버 처리용량 오버 다운.

DNS query DDOS Attack 시작 이전이나 중간쯤에 막대한 트래픽을 일으키는 재밍용 1434번 포트로의 SQL Overflow를 살짝 튕겨 준다. 예상으로는 2시 이전부터 시작 되었다고 생각한다. 트래픽이 어느 정도 궤도에 오를 때쯤 DNS를 공격 했으리라.. 그리하면 인터넷 접속도 느려 질 것이고 이천만에 육박하는 (넘나?) 사용자의 절반 이상에 해당하는 KT DNS

Internet 붕괴의 기술적 원인 분석

==========

를 사용하는 사용자들은 무자비한 Refresh Attack을 KT DNS에 쏟아 부어서 완벽한 다운을 만들어 버린다. Ddos 목표는 혜화동 KT DNS 였을 것이고 목표 달성 후 침묵 했을 것이다. 다운 이후에 다른 곳으로 트래픽을 돌렸으나 성난 네티즌들의 Refresh Attack을 감당하지 못하고 연속 다운이 이루어 진다. 다운 이후에 유유히 네트워크 트래픽을 잠식하며 udp port 1434로 SQL Overflow 코드가 돌아 다닌다. 그리고 우리는 모두 SQL Overflow 웜만 가지고 모든 걸 풀어 보려고 시도 했다. 그 안에 혹여나 뭔가가 있지는 않나 해서….

DNS Ddos Attack은 이전에 root server에 사용된 Ping flooding이 아닌 직접 query를 일으키며 실행이 되고 Ddos Attack에 사용되는 Agent들은 몇 달 혹은 며칠 전부터 은밀히 해킹을 통해 root 권한을 획득한 시스템에 올린 후 DNS 다운설명에서 보게 될 방법으로 여러 개의 같은 서브넷 상의 IP (테스트 안된 사항이다. 환경 되는 분 테스트 해보시길)를 등록하여 DNS query를 DNS 서버에 전송한다. 물론 여기서 DNS 서버는 당연히 kns.kornet.net KT DNS서버를 사용했을 것이고…
Agent를 많이 깔면 깔수록 DNS query는 더더욱 증가하게 된다. trinoo 같은 동시 공격하는 Agent 형태로 수정했을 수도 있고 아니면 DNS 트래픽이 약간 가파르게 올라 갔다면 하나씩 일일이 실행 시켰을 수도 있다. 아 그리고 해킹이 아니여도 외국서버에서 직접 DNS 서버를 kns.kornet.net으로 잡게 되면 이 것도 가능할 것이다. 완벽을 기하고 자신을 숨기기 위해서 해킹을 한 후 공격을 한 것이 당연하겠지만 말이다. 이 사이에 SQL Overflow가 udp의 장점(?)인 대량 전송으로 막대한 감염 및 트래픽 로드를 가져 오는 바람에 KT DNS 사용자 들은 인터넷 주소로 연결이 늦어 지므로 Refresh를 계속 누르고 그 외의 사용자들은 기타 ISP DNS서버는 정상 이였으나 막대한 트래픽이 선점하고 있는 인터넷 라인을 타고 목적지로 갈려니 비정상적인 연결 종료가 수시로 일어나고 웹페이지가 안 뜨는 이런 일이 벌어졌을 것이다. 기타 ISP DNS 쿼리의 양도 증가한 곳을 볼 수 있는데 이 부분은 트래픽 과다로 인한 ISP 사용자의 불만 가득한 Refresh 연속 누르기에 트래픽이 증가한 것으로 추정이 된다.

이 모든 것들이 다 들어 맞지 않는가? .. 물론 이럴 수도 있다는 것이다. ^^;

만약 이랬다면 DNS Ddos Attack 이 주공이고 SQL Overflow 가 보조 였다는 가설의 결론에 도달하게 되고 우리는 며칠동안 보조만을 죽어라고 분석하고 있었던 것이다. 이 것은 전 세계에서 비슷한 업종에 종사하는 사람들 모두 다 똑같다. 세계 어느 나라든 이와 같은 공격을 받으면 우리나라의 복구 수준을 우러러 볼 것이다. 자부심 가져도 좋다.

아마도 어느 세력이 있는가 보다. 12개 이던가 하는 root server를 다운 시키려고 ping flooding을 전송한 어느 세력이나 단체가 그때는 어느 정도의 traffic이면 다운이 가능할까 해서 계산을 한 것은 아닐는지?.. 그리고 실제적인 공격의 첫 대상은 우리나라가 아니였을

Internet 붕괴의 기술적 원인 분석

==========

까? UDP 웜 404byte에 무언가가 더 있을 것으로 몇 십번을 더 들 여다 보고 생각해 봤지만 그거는 딱 그거 뿐 이였다. 모 ISP 근무자가 위험을 감수하고 전해준 감염된 서버 및 주변 서버에서의 외부 발송 패킷 체크에서 1434 와 404 란 숫자만을 무수히 확인 했으며 모 관제업체 및 모 연구소에서 살짝 알려 준 DNS traffic 증가량과 ids 및 방화벽 로그등.. 이 모든 자료들이 이와 같은 가설을 가능하게 또 본인에게만은 확신하게 만들어 주었다. 그러나 이 글을 읽고 계실 여러분은 여러분의 상상과 판단에 맡긴다.

재미난 것은 사건 발생 시간부터가 아주 의미 있다. 우리에겐 오후 2시 분명 전세계에도 동일한 udp 패킷이 잔뜩 날아 갔을 것이다. 물론 그런 것들을 분석하고 자주 메일링 리스트에 발송하는 뛰어난 고수들은 잠들 시간이 좋은 타임 아닌가?. 대충 계산 해보니 국내 25일 오후 2시 이면 Central America에서는 24일 밤 11시쯤 되는 것 같다. 그 뒤에는 접속을 하고 싶어도 접속을 못하는 상황이 지속 되어서 더욱 더 정보 입수에 늦었던 것이구.. 공격하던 세력은 이것이 마지막 테스트 였던지 토요일 오후를 선택해 주는 배려도 있었으나 평일이라면 우리나라의 피해는 기하급수로 늘었을 것이다. 아마도 다음 번에는 인정사정 봐주지 않고 덤비지 않을까 싶다. 이런 세력이 e-terrorist 가 아닐는지..

가설을 세우고 그 가설을 받쳐줄 몇 가지 자료 확보한 뒤에 생각해 보니 실로 대단한 타이밍에 삼국지의 제갈량의 꾀를 보는 듯 하여 본인의 말투 또한 변하였으니 감안해 주시기를.. 긴 글이라서 조금 질리 셨을 듯. 다음절에서는 본인이 가장 싫어하는 제안서 형식의 그림을 보게 될 것이므로 편안하게 보셔도 될 것이다.

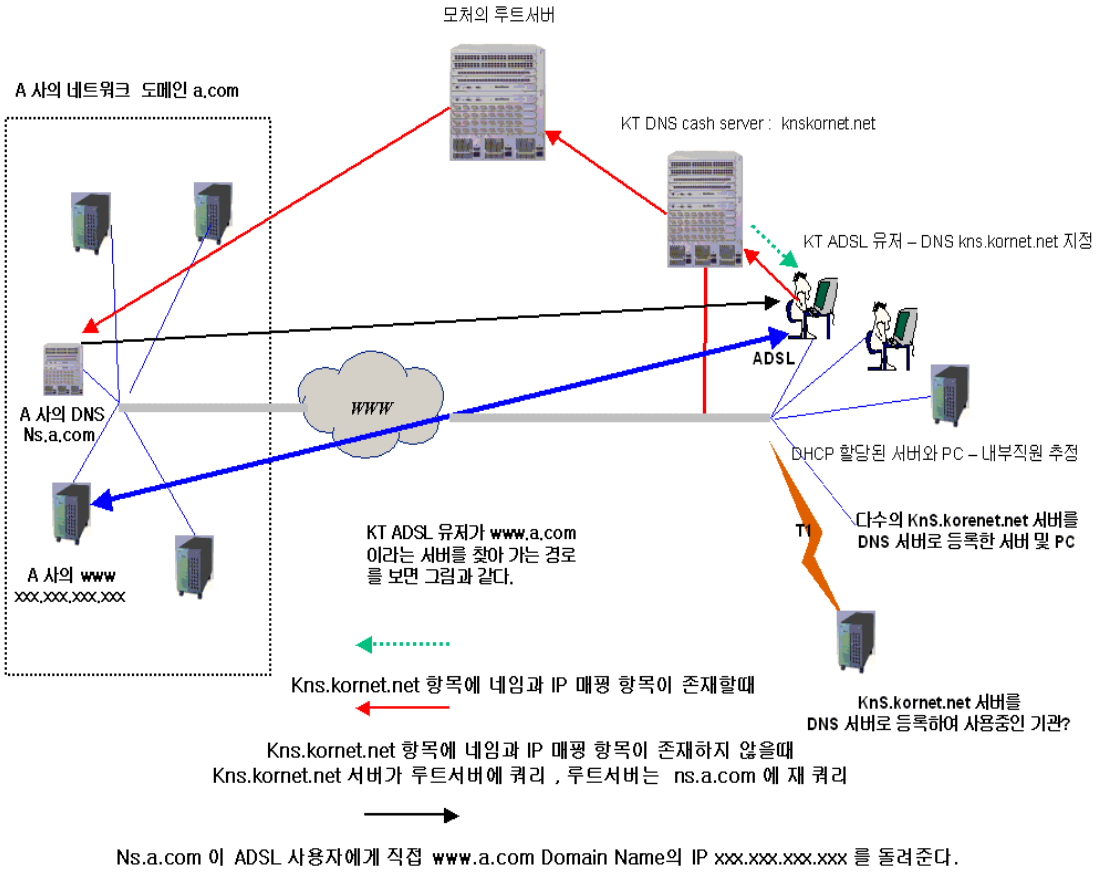
Internet 붕괴의 기술적 원인 분석

==========

I. Problem Solving

. DNS 서버는 왜 죽었는가?

먼저 DNS 가 뭔가?
* 그림 그리느라 힘들었다.



DNS 란 말 그대로 Domain Name System 이다. 그리고 이게 존재하는 이유는 IP Address 를 기억하기 보다는 연관된 이름을 기억하는 것이 더 편리하기 때문일꺼다. 매번 IP Address를 기억하고 쳐야 한다면 아주 끔찍할 것이다. www.securityindepth.net , www.securitymap.net 을 이름이 아닌 IP로 접속 해야 한다면 정말 갈 곳 몇 곳 안될 것이다. 이런 이름과 IP를 매핑 시켜 주는 시스템이 DNS 서버가 되겠다. DNS 서버의 목적은 편리함을 위해서 존재하는 것이지 DNS 서버 자체가 없다고 해도 인터넷망은 지장이 없다.

Internet 붕괴의 기술적 원인 분석

==========

다만 접속 시도를 할 때 IP주소를 일일이 다 외우거나 알고 있어야 하는 인간의 불편함만 따를 뿐.

KT DNS 가 kr을 관리하는 국가 도메인 인줄 알고 있었는데 그게 아니라는 사실을 확인해 준 동지들에게 감사 드린다. 여기에서 KT의 ADSL 사용자의 DNS 와 DHCP로 KT 라인을 사용하는 곳은 아마도 DNS가 kns.kornet.net(요거 IP가 168.126.63.1)으로 잡힐 것이다. 그리고 동지들의 말로는 많이 알려진 곳이라 다수의 서버들이 DNS로 설정하는 곳이라 했다. 그렇다면 이 kns.kornet.net DNS 서버는 아주 다수의 사용자들이 이용하는 DNS 서버 될 것이다.

(틀린 부분 있어도 넘어 가시길.. 어차피 정확한 지식을 얻을 수 없는 개인의 입장이므로.. 또한 그렇다 해도 가설은 변하지 않으므로)

왜 죽었는가? UNIX BIND를 쓰는 KT DNS에 초당 Query가 10배 증가한 것을 어떻게 해석하느냐는 질문을 받았는데 처음에는 단순한 DDos 공격에 의해 죽은 것으로 판단을 했었다. 지난 해 10.23일의 root server에 대한 ping flooding도 있었지 않은가? 그래서 추정으로 그렇게 여겼었다. 그러나 단순 DDos 공격이 아닌 DNS 쿼리가 순식간에 10배 가량 증가해서 서버가 처리 업무를 다 처리하지 못하여서 과부하에 의해 죽었다는 것을 알게 되었다. 그러면 이것은 어떤 문제로 죽었을까?

가설 그대로이다.

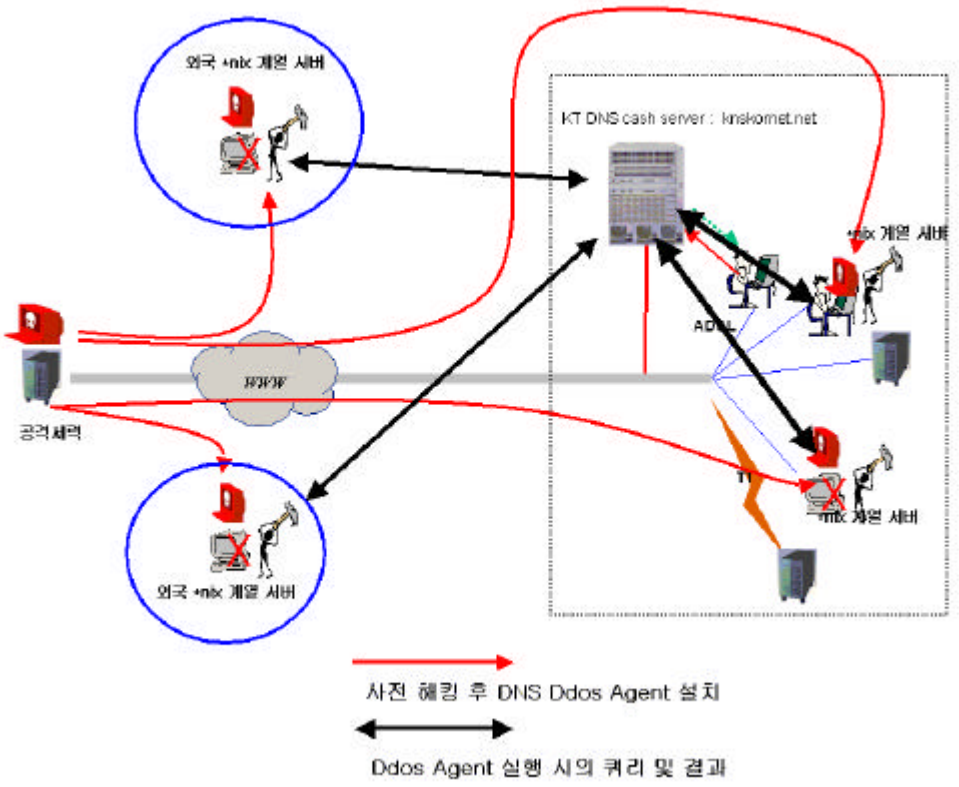
다수의 DNS query Ddos Attack -✍ KT DNS -✍ DNS 처리 용량 증가 --> KT ADSL 사용자 및 KT DNS 사용자 인터넷 접속 과정에서 느림을 인지 (domain name을 IP로 변환하는 과정이 느린거다 인터넷 라인은 무사했다.) ✍ DNS 사용자 들의 refresh 과다 반복 (DNS Query의 급격한 증가) ✍ DNS 서버 처리용량 오버 다운.

이 사이 끼어 들었거나 혹은 미리 끼어 들었을 SQL Overflow웜에 의한 데이터 트래픽도 보조로서의 공격에 대단한 공을 세웠다.

Internet 붕괴의 기술적 원인 분석

==========

그림으로 Ddos Attack의 절차 및 과정을 보도록 하자.



1.우선 먼저 사전에 kns.kornet.net DNS 서버 사용 범위에 빠른 네트웍을 이용한 공격을 하도록 하기 위해 *nix 계열 서버를 해킹 하여 root를 획득 한다.

2.더불어 미끼로 쓰기 위한 (?) 외국의 서버들도 해킹 한 후 속도가 느리긴 해도 DNS로 KT DNS를 사용하도록 설정한다.

3. 공격 시점을 기다리다가 시그널이 오면 KT DNS에 대해 DNS Ddos Attack을 수행한다.

자 그럼 이런 툴이 있을까? 어제 급히 검색을 해본 결과 몇 개 정도가 있었는데 가장 효과가 높을 것으로 판단 되는 것을 골랐다. (효과는 테스트 해본 효과가 아니다. 짐작으로 보는 효과)

공격 파일 명은 XXXXf.tar.gz (네 글자는 X로 대체한다. 검색엔진에서 조금만 찾아도 됨)
검색엔진에서 Ddos ,DNS, tool 뭐 이정도만 가지고 검색을 해도 찾을 수 있다. 물론 소스까

Internet 붕괴의 기술적 원인 분석

==========
지 공개된 형태라 코딩 능력이 있다면 인위적으로 변경이 가능하다.

Readme 만 읽어도 어느 정도 가늠할 수 있다. 천천히 보도록 하자.

What is this?

A powerful attack against DNS servers. DDNSF will send lots of queries to disable DNS.
If DNS daemon software logs incorrect queries (like most of do), affect will be greater.
If not you are still able to disable DNS.
Written and tested under Linux but should run under other unixes.

만약 부적절한 쿼리를 발생 시켜서 로그 사이즈가 커지게 되면 효고는 더 커진다고 친절하게 설명하고 있다.

이 파일은 Clent와 Server로 구성 되어 있으며 Agent역할은 다른 BO나 넷버스와 동일하게 Server 프로그램이 담당한다.
그렇다면 사용법은?

How to use?

First of all, compile files like this:

gcc -o XXXXX_client XXXXX_client.c
gcc -o XXXXX_server XXXXX_server.c

Install "XXXXX_server" on owned box. (You must be root to run XXXXX_server because of raw sockets.)
when you root on that box run program:
XXXXX_server &

XXXXX will mask itself as something like "vi" to avoid process list.

Install XXXXX_server to as many hosts as you can. More owned bandwith more power...

After all, easy and funny part comes:

Internet 붕괴의 기술적 원인 분석

==========

Write owned box's IPs to a file something like this:
192.168.5.2
192.168.5.3
192.168.5.4

lets save this as "zombie.txt"

Now run XXXXX_client:
Lets take down "www.victim_domain.com"'s DNS server.

To get DNS of a domain, type:
host -t ns www.victim_domain.com

We get DNS server 192.168.5.1

./XXXXX_client start zombie.txt 192.168.5.1

XXXXX_client will send a start command to zombies to take down victim.

To stop attack, replace start with stop.
./XXXXX_client stop zombie.txt 192.168.5.1
Use this command a few times because zombies may be a little bit busy to receive your packet.

자 사용법이 다 나왔다. 딱 여기까지만 하도록 하자. 더 길게 설명해 봐야 손 아프다.

XXXXX는 다른 문자다 오해 없으시길. 임의로 대체했다.
클라이언트와 서버를 컴파일 한 후 크래킹을 통해 root권한을 획득한 시스템에 대해 Server 프로그램을 백그라운드 프로세스로 실행을 한다. 친절하게도 vi와 같은 탐지가 어려운 형태로 실행이 된다. 크래킹을 통해 root권한을 획득하고 Server 프로그램을 많이 깔수록 효과는 더욱 커질 것이다. 만약 DNS가 IP영역에 대한 블록킹을 수행한다면 DNS의 해당영역 IP에 해당하는 서버를 해킹 하면 되는 것이고 블록킹을 수행하지 않는다면 임의의 IP에 DNS만 맞추어 주면 될 것이다.

실행은 원격에서 하나의 파일에 크래킹 된 Server가 구동중인 IP를 전부 담아 두고 목표

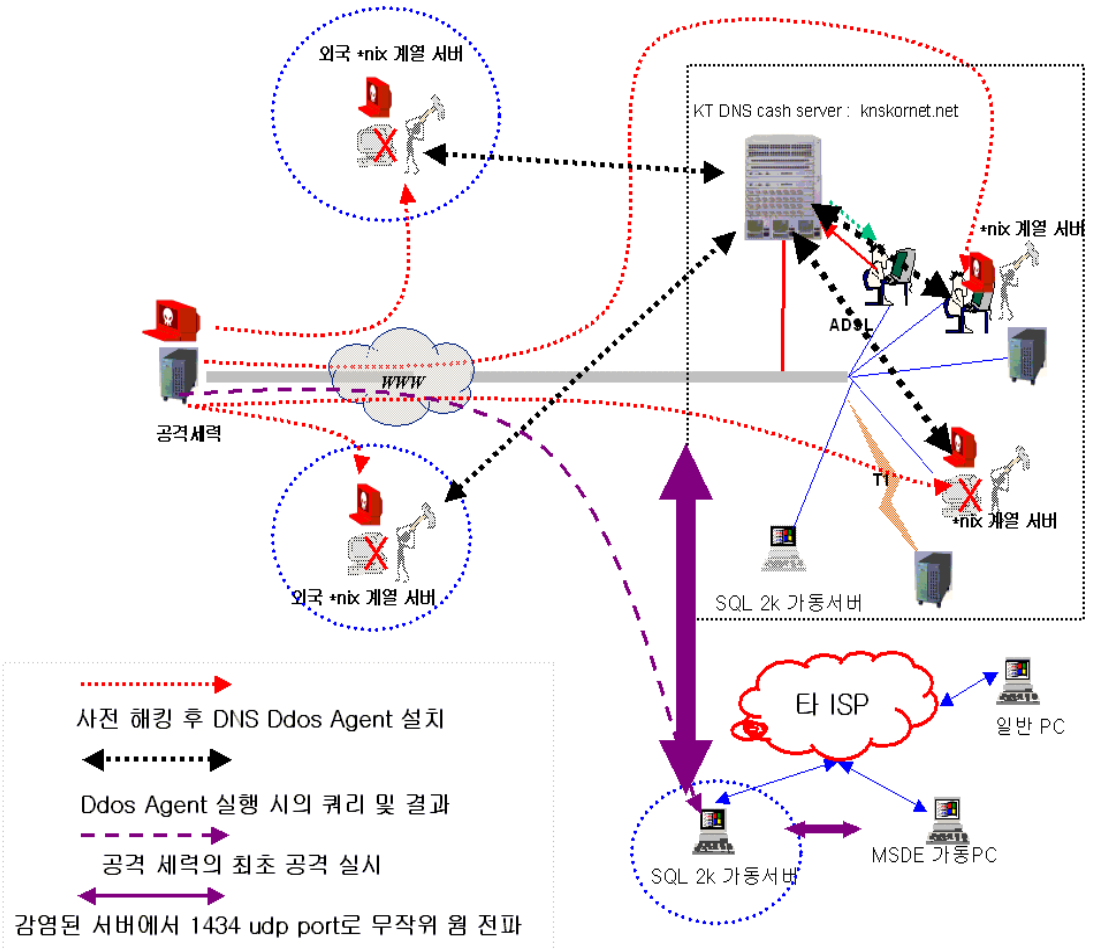
Internet 붕괴의 기술적 원인 분석

==========

DNS IP설정 ..(물론 kns.kornet.net) start 명령만 입력 후에 실행만 시키면 된다. 충분한 효과가 있으리라 생각 되는데 이 부분은 어느 정도의 로드를 동반하는지 테스트해 볼 필요는 있을 것이다. 이런 형태의 DNS에 대한 Ddos루틴이 구현이 되며 기능보강을 직접 할 수도 있을 것이다. (테스트 해보고 싶지만 환경이 허락 하지 않는다. 다만 가설의 근거 확보에 만족한다.)

*서버 기능에서 IP Spoofing 기능도 존재하며 이 기능은 Server 루틴에서 하드코딩된 부분의 IP Address를 Ddos Server가 설치된 machine의 ip address로 변경하면 된다.

주공과 보조 공격이 포함된 그림을 보자. 머리 속으로는 깨끗하게 정리가 되었는데 그림 그리다 보니 어영부영하게 되어 버렸다. 가설을 깨끗이 이해하고 공감 하시는 분 있으시면 다시 그려도 좋을 것이다.



Internet 붕괴의 기술적 원인 분석

==========

공격을 시나리오대로 풀어하면

1. 사전 작업으로 DNS Ddos 공격을 위해 해킹을 통해 root 획득 및 Agent(Server)설치

2. DNS 공격하기 얼마 전 SQL overflow worm 유포

3. 트래픽 어느 정도 발생시나 약속된 시간에 Ddos Client에서 Ddos 공격하는 명령입력

4. Agent 설치된 Server (수십? 수백?)에서 KT DNS에 대한 정상적인 DNS query를 쏟아내기 시작함

5. SQL Overflow 웜의 빠른 전파와 UDP의 특성상 패킷 하나의 dropping에 부하가 걸리는 이유로 네트웍 트래픽에 정체가 늘어남

6. 네트웍이 점차 느려짐 (KT DNS 사용자들의 과도한 Refresh 설정으로 KT DNS에 DNS query량 급증) . KT DNS를 사용하지 않는 유저들도 트래픽 폭주로 인하여 접속의 잦은 끊김과 연결 안됨을 느끼고 마찬가지로 과도한 Refresh 설정 -> 타 ISP의 DNS 서버 트래픽량 증가.

7. DDOS로 인한 DNS query양의 급격한 증가와 Refresh로 인한 DNS query 급증으로 인해 KT DNS (kns.korenet.net) 버티지 못하고 다운됨 (이 때 DNS로 KT DNS를 사용하는 유저들도 IP로 인터넷에 아주 힘들게 접속이 가능했으며 인터넷이 완전히 마비된 상태는 아니다. 그러나 아주 쉽게 인터넷 전면마비로 인식되어짐)

8. Ddos 공격은 혜화의 KT DNS로만 목표로 했을 것이기 때문에 구로나 다른 곳의 DNS가 연쇄로 다운 된 것은 처리되지 않은 다수 사용자들의 엄청난 DNS query 증가를 감당하지 못하고 다운됨.

9. 그리고 마지막 한 가지 .. 그날 이후에 생긴 DNS traffic 증가는 뉴스상으로 본 7~8시와 9시 비율이 현격히 차이 나는 건 엄청난 스트레스를 겪으며 네트웍 구조가 흔들린 이 후 여러 사용자들의 접속요구를 아직도 제대로 처리하지 못하는 경우가 있어서 약 트래픽이 세배 정도 된다고 했는데 일 인당 세 번 정도의 refresh 요청을 더 했다고 생각하면 편할 것 같다.

Internet 붕괴의 기술적 원인 분석

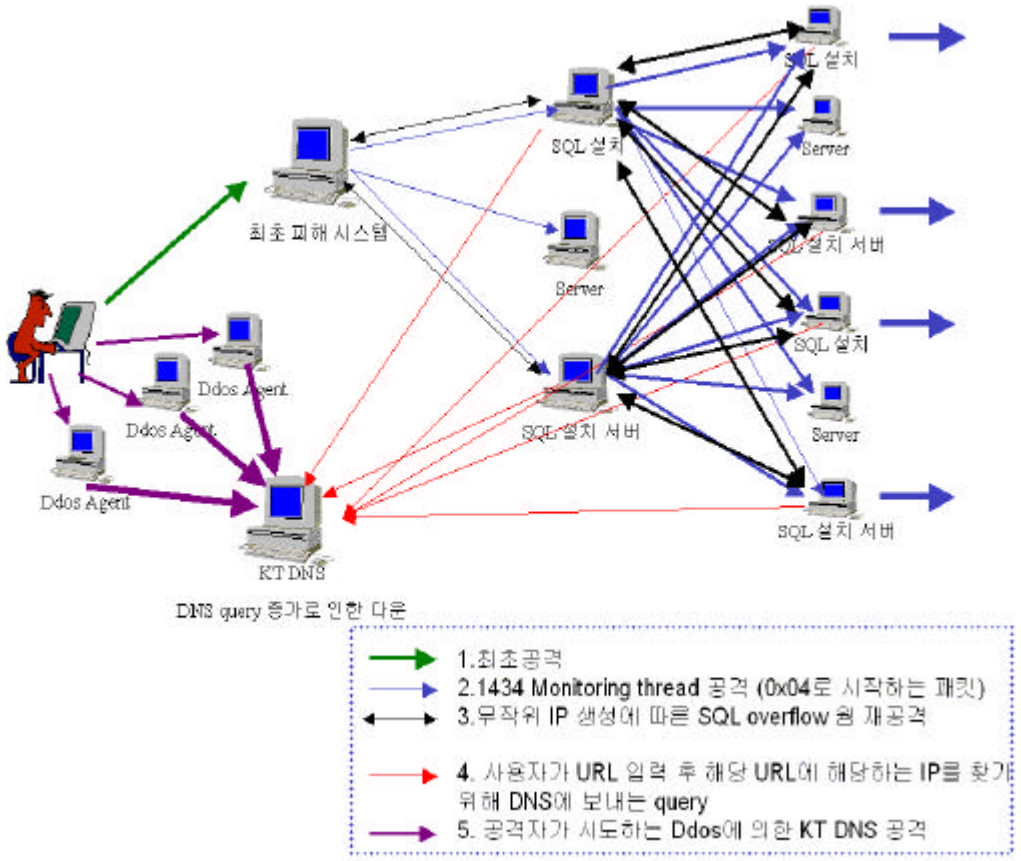
==========

즉 정리해 보면 트래픽에 흘러간 공격 패턴은 행위에 따라 딱 두 가지다.

1). 보조공격수: 1434 Monitoring port를 직접 공격한 패턴 (주된 감염 시도 패턴)
(버퍼 오버런을 일으켜서 SQL Server가 운영되는 System 레벨의 권한에서 임의의 명령을 실행할 수 있는 능력을 가지게 된다.)
First byte : 0x04로 시작하는 패턴 ,임의의 IP Address에 무작위 발송

2) 주 공격수: KT DNS를 직접 DNS Ddos 공격하는 정상적인 형태의 DNS query (패턴이라고 하기에도 뭐하다.) DNS에게 가해지는 query 폭탄이라고 하는 게 나을 듯.

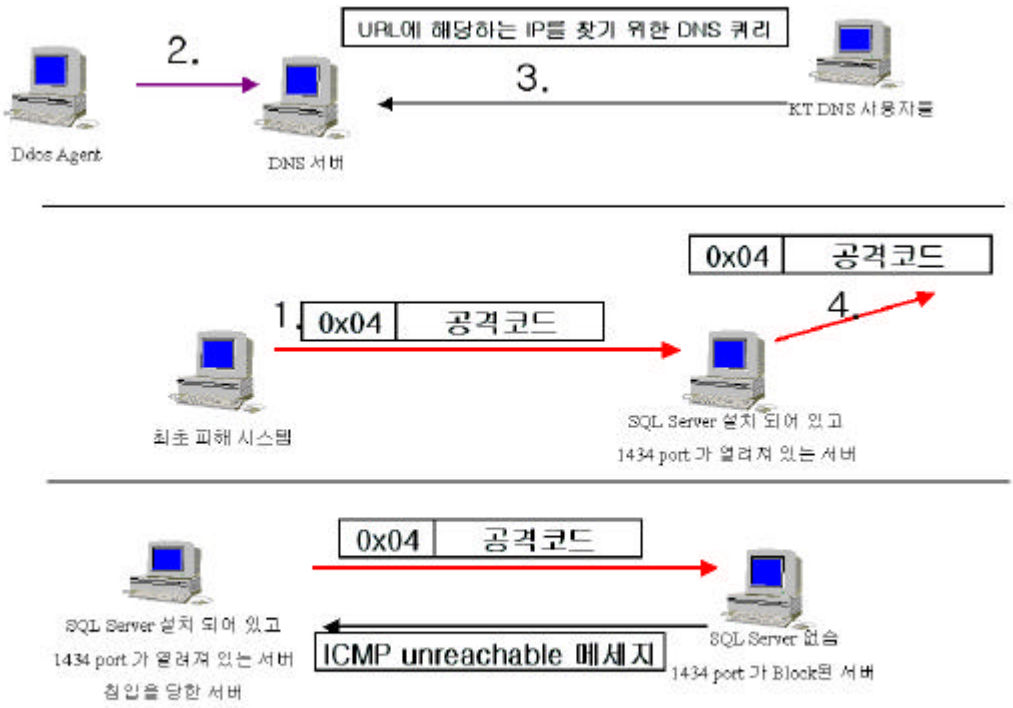
이 두 공격수에 의해 또 좋은 인프라와 인구대비 컴퓨터 사용 비율이 높은 이유로 주말임에도 불구하고 기반망의 붕괴가 연쇄적으로 이루어 졌음을 개인적인 조사 및 예상에 의해 판단하는 바이다. 이게 끝이다.



Internet 붕괴의 기술적 원인 분석

==========

아래 이미지의 순서는 실행 되는 순서를 예측한 것임



현재까지 국내외로 조사된 결과로는 웜의 전파방식과 코드에 대한 disassemble 루틴이 알려져 있으며 (이게 전부다. 왜냐면 보조 공격수이기 때문에) 전송 코드에 대한 자세한 디스어셈블 루틴은 http://www.techie.hopto.org/sqlworm.html 여기를 참고 하기 바란다.

본인은 이 가설 이외에는 설명할 수 있는 것이 없다고 판단 하는데 다른 의견도 있을 수 있을 것이다. 중요한 점은 본인이 생각한 가설도 바로 적용이 가능하다는 것에 있다. 정확한 사건에 대한 Forensic을 할 수는 없겠지만 본인이 생각한 가설도 동일한 결과를 가져 올 것은 분명하다. 그 이외의 모든 시도는 아마 좀 힘들지 않을까? 그 이유와 이전에 주장한 본인의 가설에 대해서는 타당하지 않음을 위에서 언급을 다 했다. 충분한 설명이 될 것이다. 조금만 더 생각한다면...

Internet 붕괴의 기술적 원인 분석

==========

II. Technical Analysis

1월 26일 올라온 eeye사의 공격코드에 대한 disassembly 루틴과 버그트랙 메일링 리스트에 올라온 여러 정보들을 종합해 보도록 하자.

생략..

...

entrypoint

```
xor     ecx, ecx
push    ecx
push    ecx
push    eax
xor     ecx, 9B040103h
xor     ecx, 1010101h
push    ecx             ; 9A050002 = port 1434 / AF_INET
lea     eax, [ebp-34h]  ; (socket)
push    eax
mov     eax, [ebp-40h]  ; ws2_32 base address
push    eax
call    dword ptr [esi] ; GetProcAddress
push    11h
```

....

이 부분은 1434 포트에 Keep alive packet을 보내는 과정으로 추정이 된다. 1434 port에 keep alive packet을 전송하는 코드의 일부분으로 보여진다.

PRND:

```
mov     eax, [ebp-4Ch]  ; Pseudo Random Algorithm Start
lea     ecx, [eax+eax*2]
lea     edx, [eax+ecx*4]
shl     edx, 4
add     edx, eax
shl     edx, 8
sub     edx, eax
lea     eax, [eax+edx*4]
```

Internet 붕괴의 기술적 원인 분석

```
================================================
            add     eax, ebx        ; Pseudo Random Algorithm End
            mov     [ebp-4Ch], eax
            push    10h
            lea     eax, [ebp-50h]
            push    eax
            xor     ecx, ecx
            push    ecx
            xor     cx, 178h
            push    ecx
            lea     eax, [ebp+3]
            push    eax
            mov     eax, [ebp-54h]
            push    eax
            call    esi             ; sendto
           jmp     short PRND      ; Jump back to Pseudo Random Algorithm Start
```

이 루틴은 임의의 주소루틴을 생성하고 전송하는 루틴이다. 앞부분의 생략된 부분에서는 Reverse Packet 방식에 따라 Worm과 EIP 그리고 Garbage 데이터의 Address를 변경하는 루틴이 들어 있다.

좀 더 많은 시간과 여유가 주어 진다면 정밀하게 분석해 보고 싶지만 허락 하지 않는 환경이 있다.

간단히 코드에 대한 분석을 했었고 여러 정황에 대한 판단 근거를 가지고 있다는 사실만으로 Technical 부분은 마치고자 한다. 근 일주일 가량 폐인처럼 살고 있다. 아..X같다.

III. 결론

결론은 이전 문서와 다르다..당연히...

주공격수와 보조공격수의 업무가 확실히 분담되어 있었으며 보조 공격수의 선택을 탁월하게 한 관계로 주공격수의 임무를 눈치 못 채도록 하는데 절반의 성공을 거두었다고 생각한다. (왜 절반이냐?.. 의심을 가지고 있는 모든 동지 분들과 본인은 알아 채지 않았는가?. ㅎㅎ.. 뭐 아님 어떤가? 본인이야 그냥 개인일 뿐이니 쪽 한번 팔리면 끝이다.)

우리는 왜 주공을 알아채지 못했을까?.. 정상적인 DNS query였기 때문일 것이다. 분명 본인이 제시한 가설대로라면 25일 오후쯤에 그 수많은 쿼리에서도 반복해서 루프를 도는 주소들이 섞여 있을 것이다. 엄청난 데이터 속에서 찾는 것이 힘들겠지만. 또 찾았다 해도 앞전에 이야기 한대로 분명 해킹 한 시스템에 Agent를 깔아서 공격 했을 터인데 어떻게 잡을

Internet 붕괴의 기술적 원인 분석

==========

수 있는가? 지난 해 root server에 대한 ping flooding 공격 때에도 NIPC(국가기반시설 보호 센터) , FBI,CIA 뭐 하나 더 해서 NSA 까지 잡는다고 한 것 같은데 아직 소식 없다. 우리도 마찬가지 이다. 이미 지나간 일이다. 잡을 수도 없을 뿐더러 아주 악의적인 목적이였기에 더더욱 찾기 힘들게 숨었을 것이다.
이번 공격에 사용된 방식과 형식은 한 개인이 행동하기에는 너무 규모가 크다. 최소 몇인 이상이 일정기간의 준비기간을 두고 차근히 준비해 왔을 것이다. 9.11 테러에서 5년 전부터 비행군사학교에서 뱅기 모는 훈련을 받으면서 준비한 테러리스트 처럼...이들은 e-terrorist 들이다. 조만간 사이버 무기로 협박하는 테러리스트들을 자주 보게 될지도.. 우리는 처음으로 대규모 사이버 공격의 현장을 목격하고 경험한 나라의 사람들이다. 인프라의 보호에 대해 독자적인 방법과 고민이 앞으로 이루어 져야 할 것이다.

주공을 모르고 보조만을 보고 별의 별 가능성 다 고려하다 보니 억지스런 가설도 나오게 되었고 그런 점으로 인해 며칠간의 양심의 찔림이 계속 있었다. 이제 서야 조금 홀 가분 해진 것 같다. 엉터리라도 본인 나름대로는 마음 편하기 때문이겠지만..

1.25일 본인의 헛일 내역에 대해 잠시 살펴보자

✍✍ .토요 휴무일: 11시에 약속이 있어 나감 . 11시 30분에 점심식사.
✍✍ .직장에서 밀린 문제 해결 재촉 전화 옴 .
✍✍ .1시 30분에 귀가 후 이메일로 문제해결 시도 중..네트웍이 잘 안되는 상황 발생
✍✍ .이메일 계정 5개정도를 사용해서 아웃룩에서 다운 받는데 전부 특이한 형태를 보임
✍✍ (특이한 형태 : 연결은 맺어지나 다운 받는 도중 혹은 연결되자 마자 TCP 연결이 리셋됨)
✍✍ 겨우 겨우 문제 해결 메일 발송 (진짜 운 좋았음)
✍✍ .2시 무렵 MSN 로그인 (너무 한적함을 느낌. 서른 명 가까운 리스트에 단 한명)
✍✍ . 그 단 한명이랑 채팅 (보안관제 분야에 근무하는 후배)
✍✍ . 잦은 접속 중단으로 네트웍 상태 확인하다 이상한 점 발견 send byte/received byte 비율이 비정상적 형태 5:1 정도의 비율이였음. 특이한 형태의 프로그램을 실행 하지 않았는데도 불구하고 비율이 너무 차이남.
✍✍ 메일서버 연결이 안되어 traceroute 했으나 특이한 반응에 뭔가 문제가 있다고 생각함
✍✍ 대화 하던 동생넘이 일 생겼다고 나갈라고 함. 잠시 예의상 무슨 일이냐고 물음 . IDC 네트웍이 자꾸 죽는다고 함.
✍✍ 그 뒤에 이것 저것 알아보고 여기저기 연락. 이게 시작이다.

눈치 빠른 사람들은 행적을 보며 대충 감 잡았으리라 본다. 나도 우연히 발견했다. ㅎㅎ

Internet 붕괴의 기술적 원인 분석

==========

자 그럼 대책은?

대책까지 여기서 왈가왈부 할 필요 없다고 생각한다. 우리나라의 똑똑한 사람들 무지 많으니 문제점만 보고도 턱 하니 대책이 나올 것이다. 좀 더 넓은 시야의 대책이 이번 사건으로 인해 준비되지 않을까 기대해 본다.

여백

본인의 욕심과 과욕으로 인하여 지난 주말부터 엉망인 생활에 원망을 가득 품고 상당히 삐쳐있는 집사람에게 용서를 빌어본다. (T,T. 이거 해야 설날 산너머 먼데 있는 고향 같이 간다.) 딱 그거 밖에 없는 본인에게 시집와서 몸과 마음 고생 시킨 점 긴 글 빌어 감사의 마음 전한다.

동틀 무렵까지 이거 쓰다가 잠이 들어 회사도 지각했다. 3일 동안 무시하고 지각까지 하고 짤리는 수도 있을 수 있다. 가장으로서 험난한 생활로 돌아가 세금 고민하고 매달을 걱정하는 삶이 기다린다. 얼른 돌아가서 진정 시켜야 살아갈 수 있다.

그러나 이런 일 아니면 언제 청춘의 열정을 불 사를 수 있겠나? 아무 대가 없지만 후회 없이 스스로에게 만족하므로 기쁘다. 여러 곳에서 짧은 지식의 본인을 가르쳐 주시고 아낌없이 알려준 동지들이 도움을 주신 것에 대해서는 이 문서로 감사의 인사를 대신 한다.

제시한 문제점과 지적이 틀렸을 경우엔 지체하지 말고 연락 주시기 바란다. 사소한 문제나 철자 이런 건 그냥 넘어가자. 근본적인 가설에 문제가 있을 경우는 수정을 요구하시라 그래야 명랑 사회가 유지 될 수 있으니 말이다.

즐겁고 신나는 한해의 액땜으로 생각하고 문제점을 정비하길 기원해 본다. 액땜이 컸으니 모두에게 골고루 돌아가는 큰 복이 반드시 있을 것이다.

Ps: 다음 번에는 중국 분석 이랑 기반시설 보호에 대해서 한번 더 문서를 만들어 보겠다. 언제나 그렇듯이 하고 싶은 맘이 들면 바로 한다. 하지만 한동안은 힘들 듯 너무 엑기스를 많이 썼다. 바람 불면 부는 대로 ~~ 살랑..

연락처: winsnort@hotmail.com (MSN)

Email: winsnort@securityindepth.net , winsnort@skinfosec.co.kr

--

소심하고 안정적인 상태의 성공을 바라기 보다는

거대하고 야심찬 꿈의 실현에서의 실패를

나는 바라노라. - 절망의 바다에서

--