Using Network−Based Application Recognition and ACLs for Blocking the "Code Red" Worm

Document ID: 27842

Contents

Introduction
Prerequisites
Requirements
Components Used
Conventions
How to Block the "Code Red" Worm
Supported Platforms
Detect the Infection Attempt in the IIS Web Logs
Mark Inbound "Code Red" Hacks Using IOS Class−Based Marking Feature
Method A: Use an ACL

Method B: Use Policy−Based Routing (PBR)
Method C: Use Class−Based Policing
NBAR Restrictions
Known Issues
Related Information

Introduction

This document provides a method for blocking the "Code Red" worm at network ingress points through Network−Based Application Recognition (NBAR) and Access Control Lists (ACLs) within Cisco IOS® Software on Cisco routers. This solution should be used in conjunction with the recommended patches for IIS servers from Microsoft.

Note: This method does not work on Cisco 1600 series routers.

Note: Some P2P traffic cannot be completely blocked due to the nature of its P2P protocol. These P2P protocols dynamically change their signatures to bypass any DPI engines trying to completely block their traffic. Therefore, it is recommended to limit the bandwidth instead of completely blocking them. Throttle the bandwidth for this traffic. Give much less bandwidth; however, let the connection go through.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Quality of Service (QoS) service policies using the commands of the modular QoS command line interface (CLI).
- NBAR
- ACLs
- Policy−based routing

Components Used

This document is not restricted to specific software and hardware versions. The configuration in this document was tested on the Cisco 3640 that runs Cisco IOS version 12.2(24a)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

How to Block the "Code Red" Worm

The first thing you should do to combat "Code Red" is apply the patch available from Microsoft (see links in section Method A: Use an ACL below). This protects vulnerable systems and removes the worm from an infected system. However, applying the patch to your servers only prevents the worm from infecting the servers, it does not stop the HTTP GET requests from hitting the servers. There is still the potential for the server to get bombarded with a flood of infection attempts.

The solution detailed in this advisory is designed to work in conjunction with the Microsoft patch to block the "Code Red" HTTP GET requests at a network ingress point.

This solution attempts to block the infection, however it will not cure problems caused by the buildup of large numbers of cache entries, adjacencies, and NAT/PAT entries, since the only way to analyze the contents of the HTTP GET request is following the establishment of a TCP connection. The following procedure will not help protect against a scan of the network. However, it will protect a site from infestation from an external network or reduce the number of infection attempts that a machine must service. In combination with inbound filtering, outbound filtering prevents infected clients from spreading the "Code Red" worm to the global Internet.

Supported Platforms

The solution described in this document requires the class−based marking feature within Cisco IOS software. Specifically, the ability to match on any part of an HTTP URL uses the HTTP sub−port classification feature within NBAR. The supported platforms and minimum Cisco IOS software requirements are summarized below:

Platform	Minimum Cisco IOS Software
7200	12.1(5)T
7100	12.1(5)T
3745	12.2(8)T
3725	12.2(8)T
3660	12.1(5)T
3640	12.1(5)T
3620	12.1(5)T
2600	12.1(5)T

1700	12.2(2)T

Note: You need to enable Cisco Express Forwarding (CEF) in order to use NBAR.

Class−based marking and Distributed NBAR (DNBAR) are also available on the following platforms:

Platform	Minimum Cisco IOS Software
7500	12.1(6)E
FlexWAN	12.1(6)E

Detect the Infection Attempt in the IIS Web Logs

The initial infection attempt sends a large HTTP GET request to the target IIS server. The original "Code Red" footprint is shown below:

```
2001−08−04 16:32:23 10.101.17.216 − 10.1.1.75 80 GET /default.ida
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u
7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a 403
```

The "Code Red" II footprint is shown below:

```
2001−08−04 15:57:35 10.7.35.92 − 10.1.1.75 80 GET /default.ida XXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX%u9090
%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%
u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a 403 −
```

Notice that the GET request is always looking for a file with an .ida extension. This is a common string in all infection attempts and can therefore be used as a match criteria with class−based marking in IOS. The remainder of the GET request will not necessarily be consistent as it is just attempting to create a buffer overflow. This can be seen by comparing the two entries above.

It is now being reported that the difference between these two signatures is due to a new strain of the "Code Red" worm, dubbed CodeRed.v3 or CodeRed.C. The original "Code Red" strain contains the "NNNNNNNN" string in the GET request, while the new strain contains "XXXXXXXX". Refer to the Symantec Advisory for more details.

At 6:24PM EDT, 6 Aug 2001, we recorded a new footprint. We have since learned that this is the footprint that is left behind by the eEye vulnerability scanner .

```
2001−08−06 22:24:02 10.30.203.202 − 10.1.1.9 80 GET /x.ida AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAX 403 HTTP/1.1 −
```

The technique for blocking "Code Red" provided in this advisory can also block these scanning attempts simply by tightening the class map definition as shown in the next section.

Mark Inbound "Code Red" Hacks Using IOS Class−Based Marking Feature

To block the "Code Red" worm, use one of the three methods described below. All three methods classify malicious traffic using the Cisco IOS MQC feature. This traffic is then dropped as described below.

Method A: Use an ACL

This method uses an ACL on the output interface to drop the marked "Code Red" packets. Let's use the following network diagram to illustrate the steps in this method:

Here are the steps to configuring this method:

1. Classify inbound "Code Red" hacks with the class−based marking feature in Cisco IOS software, as shown below:

   ```
   Router(config)#class-map match-any http-hacks
   Router(config-cmap)#match protocol http url "*default.ida*"
   Router(config-cmap)#match protocol http url "*cmd.exe*"
   Router(config-cmap)#match protocol http url "*root.exe*"
   ```

   The above class map looks inside of HTTP URLs and matches any of the specified strings. Notice that we have included other file names besides the default.ida of "Code Red". You can use this technique to block similar hack attempts, such as the Sadmind virus, which is explained in the following documents:

   - http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00−078.asp
   - http://www.sophos.com/virusinfo/analyses/unixsadmind.html
2. Build a policy and use the **set** command to mark inbound "Code Red" hacks with a policy map. This document uses a DSCP value of 1 (in decimal) since it is unlikely that any other network traffic is carrying this value.

   Here we mark inbound "Code Red" hacks with a policy map named "mark−inbound−http−hacks".

   ```
   Router(config)#policy-map mark-inbound-http-hacks
   Router(config-pmap)#class http-hacks
   Router(config-pmap-c)#set ip dscp 1
   ```
3. Apply the policy as an inbound policy on the input interface to mark arriving "Code Red" packets.

   ```
   Router(config)#interface serial 0/0
   Router(config-if)#service-policy input mark-inbound-http-hacks
   ```
4. Configure an ACL that matches on the DSCP value of 1, as set by the service policy.

   ```
   Router(config)#access-list 105 deny ip any any dscp 1
   Router(config)#access-list 105 permit ip any any
   ```

   Note: Cisco IOS Software Releases 12.2(11) and 12.2(11)T introduce support for the **log** keyword on the ACL in defining on class maps for use with NBAR (CSCdv48172). If you are using an earlier

release, do not use the **log** keyword on the ACL. Doing so forces all packets to be process−switched instead of CEF−switched, and NBAR will not work since it requires CEF.

5. Apply the ACL outbound on the output interface that connects to the target Web servers.

```
Router(config)#interface ethernet 0/1
Router(config-if)#ip access-group 105 out
```

6. Verify that your solution works as expected. Execute the **show access−list** command and ensure that the "matches" value for the deny statement is incrementing.

```
Router#show access-list 105
Extended IP access list 105
 deny ip any any dscp 1 log (2406 matches)
 permit ip any any (731764 matches)
```

In the configuration step, you can also disable sending IP unreachable messages with the **no ip unreachable** interface−level command to avoid causing the router to expend excessive resources.

This method is not recommended if you can policy−route the DSCP=1 traffic to Null 0, as described in the Method B section.

Method B: Use Policy−Based Routing (PBR)

This method uses policy−based routing to block marked "Code Red" packets. You do not need to apply the commands in this method if methods A or C are already configured.

Here are the steps to implementing this method:

1. Classify the traffic and mark it. Use the **class−map** and **policy−map** commands shown in method A.
2. Use the **service−policy** command to apply the policy as an inbound policy on the input interface to mark arriving "Code Red" packets. See method A.
3. Create an extended IP ACL that matches on the marked "Code Red" packets.

```
Router(config)#access-list 106 permit ip any any dscp 1
```

4. Use the **route−map** command to build a routing policy.

```
Router(config)#route-map null_policy_route 10
Router(config-route-map)#match ip address 106
Router(config-route-map)#set interface Null0
```

5. Apply the route−map to the input interface.

```
Router(config)#interface serial 0/0
Router(config-if)#ip policy route-map null_policy_route
```

6. Verify your solution works as expected with the **show access−list** command. If you are using output ACLs and have enabled ACL logging, you also can use the **show log** commands, as shown below:

```
Router#show access-list 106
Extended IP access list 106
 permit ip any any dscp 1 (1506 matches)

Router#show log
Aug 4 13:25:20: %SEC-6-IPACCESSLOGP:
 list 105 denied tcp A.B.C.D.(0) -> 10.1.1.75(0), 6 packets
```

```
Aug 4 13:26:32: %SEC-6-IPACCESSLOGP:
 list 105 denied tcp A.B.C.D.(0) -> 10.1.1.75(0), 6 packets
```

We are able to make the discard decision at the ingress interface of the router, rather than needing an output ACL on every egress interface. Again, we recommend disabling the sending IP unreachable messages with the command **no ip unreachables** command.

Method C: Use Class-Based Policing

This method generally is the most scalable as it does not depend on either PBR or output ACLs.

1. Classify the traffic using the **class-map** commands shown in method A.
2. Build a policy using the **policy-map** command and use the **police** command to specify a drop action for this traffic.

```
Router(config)#policy-map drop-inbound-http-hacks
Router(config-pmap)#class http-hacks
Router(config-pmap-c)#police 1000000 31250 31250
 conform-action drop exceed-action drop violate-action drop
```

3. Use the **service-policy** command to apply the policy as an inbound policy on the input interface to drop the "Code Red" packets.

```
Router(config)#interface serial 0/0
Router(config-if)#service-policy input drop-inbound-http-hacks
```

4. Verify that your solution works as expected with the **show policy-map interface** command. Ensure that you see incrementing values for the class and the individual match criteria.

```
Router#show policy-map interface serial 0/0

Serial0/0

  Service-policy input: drop-inbound-http-hacks

    Class-map: http-hacks (match-any)
      5 packets, 300 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol http url "*default.ida*"
        5 packets, 300 bytes
        5 minute rate 0 bps
      Match: protocol http url "*cmd.exe*"
        0 packets, 0 bytes
        5 minute rate 0 bps
      Match: protocol http url "*root.exe*"
        0 packets, 0 bytes
        5 minute rate 0 bps
      police:
        1000000 bps, 31250 limit, 31250 extended limit
        conformed 5 packets, 300 bytes; action: drop
        exceeded 0 packets, 0 bytes; action: drop
        violated 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps

    Class-map: class-default (match-any)
      5 packets, 300 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

NBAR Restrictions

When using NBAR with the methods in this document, note that the following features are not supported by NBAR:

- More than 24 concurrent URLs, HOSTs or MIME type matches
- Matching beyond the first 400 bytes in a URL
- Non−IP traffic
- Multicast and other non−CEF switching modes
- Fragmented packets
- Pipelined persistent HTTP requests
- URL/HOST/MIME/ classification with secure HTTP
- Asymmetric flows with stateful protocols
- Packets originating from or destined to the router running NBAR

You can't configure NBAR on the following logical interfaces:

- Fast EtherChannel
- Interfaces that use tunneling or encryption
- VLANs
- Dialer interfaces
- Multilink PPP

Note: NBAR is configurable on VLANs as of Cisco IOS Release 12.1(13)E, but supported in the software switching path only.

Since NBAR cannot be used to classify output traffic on a WAN link where tunneling or encryption is used, apply it instead to other interfaces on the router, such as the LAN interface, to perform input classification before the traffic is switched to the WAN link for output.

For more NBAR information, see the links in the Related Information section below.

Known Issues

The "Code Red" worm exploits a vulnerability on unpatched servers within IIS that uses the Microsoft Indexing Service. The Internet Data Administration script file (default.ida) is installed by default on all IIS servers. "Code Red" relies on the presence of this file to carry out the exploit. Most systems do not use this service so the blocking method provided in this advisory will be effective. However, some servers may use this service within IIS. In this case, the blocking method proposed here could block legitimate requests to the IIS server.

Related Information

- **Dealing with mallocfail and High CPU Utilization Resulting From the "Code Red" Worm**
- **Using Cisco Secure IDS/NetRanger Custom String Match Signatures for "Code Red" Worm Remote Buffer Overflow in Microsoft Index Server ISAPI Extension in IIS 4.0 and 5.0**
- **Technical Support & Documentation − Cisco Systems**