



SECURITY BULLETIN – Worldwide H.323 Botnet Calling H.323 Systems as “cisco”
Version 1.1

Security Bulletin Relating to Worldwide Botnet Dialing H.323-Capable Systems

DATE PUBLISHED: May 6, 2015

Effective January 1, 2014, the below applies to all Polycom security publications:

Polycom may at times issue either a **Security Advisory** or a **Security Bulletin** with regards to a particular vulnerability or set of vulnerabilities. If a **Security Advisory** is issued, this means that one more Polycom products are verified by Polycom to be affected by one or more vulnerabilities.

If a **Security Bulletin** is issued, Polycom is providing its customers with information about one or more vulnerabilities that have not been found by Polycom to directly affect any Polycom products, but that may be mistakenly thought to affect Polycom products. A **Security Bulletin** might also be issued when a customer’s environment might be affected, when false positives might occur from vulnerability scans, or when any other possible (but not actual) concern might exist regarding Polycom products and the vulnerability.

This information applies only to those Polycom products that might answer, respond to, or forward an H.323 call.

This advisory is NOT about any vulnerability in any Polycom product. This advisory serves to educate those for whom unwanted inbound H.323 calls have become a problem. Polycom products are NOT demonstrating a vulnerability when they respond properly to H.323 calls.

Situation Summary

A “botnet”, or mass group of systems on the Internet under the control of one or several malicious entities, is being used to place H.323 calls to any device on the Internet that can meaningfully respond to

an H.323 call. This means endpoints, bridges, MCU's, gatekeepers, firewalls, proxies, etc. might all be called by the botnet.

The purpose of the botnet is unclear, although toll fraud is usually the purpose behind such broad sweeps against systems that speak a specific communications protocol. The systems used in the botnet are, as is typical with this method, innocent third-party hosts whose owners are most likely unaware are a part of the botnet.

The botnet can successfully be run against all H.323 systems it finds on the Internet, and is presumably noting which systems respond to calls for future attacks.

This particular botnet advertises itself by the fact that it seems to consistently use "h323-ID = 'cisco'". The name appears to be unrelated to anything in the real world, and is rather a way of adding seeming legitimacy to the H.323 botnet.

Situation Details

The botnet appears to be at least partially (probably mostly) comprised of web servers around the Internet, all of which appear to be operating below recommended security patch levels. Such systems are owned by innocent third parties who do not patch their systems to current levels. The attackers first use known security flaws to compromise such systems, and then insert code that dials H.323 systems and reports back to the individual(s) who created this illegal system ("bot master" and "bot herder" are the normal terms for the one(s) in control of the botnet.

Every system we have been able to analyze (but one) conforms with this model. One of the IP addresses used to call H.323 systems was analyzed and is shown below. The one system that did not conform with the under-patched web server model was an instance of FreePBX – most likely compromised in a similar manner for identical purposes.

From the perspective of the Internet, this is just a random web server whose owner failed to patch things when known vulnerabilities were announced (we had permission to scan). Such systems might be behind on patching Apache or OpenSSL or other similar modules. This one (as reported by Nikto, a free vulnerability scanner) has a vulnerability from 2002 among its many vulnerabilities.

Again, this is NOT a "bad guy" system. This is a third-party web server, poorly maintained, compromised by the attacker(s), and then loaded with the software that is performing the H.323 calls:

```

- Nikto v2.1.6
-----
+ Target IP:
+ Target Hostname:
+ Target Port:      80
+ Start Time:      2014-11-05 13:07:55 (GMT-8)
-----
+ Server: Apache/2.2.14 (Unix) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.81 PHP/5.3.1 mod
_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
+ Retrieved x-powered-by header: PHP/5.3.1
+ The anti-clickjacking X-Frame-Options header is not present.
+ Root page / redirects to: http://fischer.appiaservices.com/xampp/
+ Server leaks inodes via ETags, header found with file /favicon.ico, inode: 305
42811, size: 30894, mtime: Fri May 11 05:40:36 2007
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to e
asily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d1
5. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, H
TTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_N
OT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FO
UND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.h
tml.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.v
ar, HTTP_NOT_FOUND.html.var
+ PHP/5.3.1 appears to be outdated (current is at least 5.4.28)
+ OpenSSL/0.9.81 appears to be outdated (current is at least 1.0.1e). OpenSSL 0.
9.8r is also current.
+ Apache/2.2.14 appears to be outdated (current is at least Apache/2.4.7). Apach
e 2.0.65 (final release) and 2.2.26 are also current.
+ mod_ssl/2.2.14 appears to be outdated (current is at least 2.8.31) (may depend
on server version)
+ mod_perl/2.0.4 appears to be outdated (current is at least 2.0.7)
+ mod_apreq2-20090110/2.7.1 appears to be outdated (current is at least 2.8.0)
+ Perl/v5.10.1 appears to be outdated (current is at least v5.14.2)
+ OSUDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to X
ST
+ mod_ssl/2.2.14 OpenSSL/0.9.81 PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0
4 Perl/v5.10.1 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer over
flow which may allow a remote shell. CVE-2002-0082, OSUDB-756.

```

Impact and Risk

Since the attack is so widespread, since its purpose cannot be truly uncovered, and since each target network is different both in purpose and in layout, no one statement can be made about impact and risk. As noted above, botnets that dial communication protocols over and over to as many hosts as possible are quite likely looking for a means to conduct toll fraud – dialing calls through third-party systems, without the consent of the owners of those third-party systems.

General Mitigations

The challenge with these scenarios is that the target hosts (all H.323-listening devices) are doing exactly what they are set up to do: answering calls. Polycom products are not vulnerable to anything in this situation. Mitigations to relieve the burden presented by these non-stop calls therefore come in the form of adding some ability to discern desired calls from undesired calls.

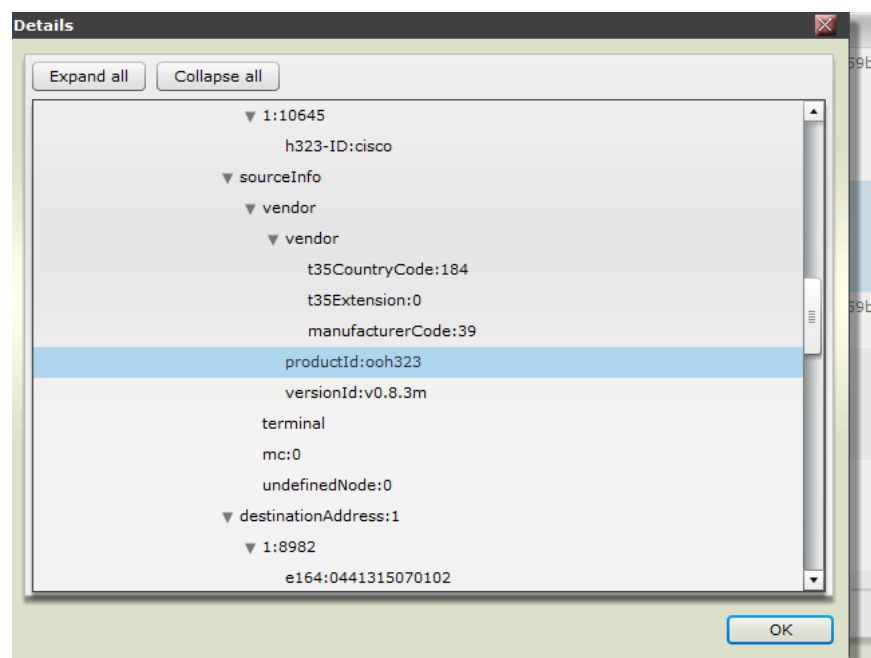
Whitelisting is the most common method for separating the good calls from the bad. In an environment where all of the “good guys” are known, it is best to simply create whitelists comprised of those addresses, and to deny all inbound calls NOT from those addresses.

Other times blacklists can be grown and maintained, blocking each of the botnet callers as it is logged and confirmed to be a “bad guy”.

Fortunately, the botnet has also proven (so far) to consistently populate two key H.323 fields with the same information.

H.323 Alias Value: (h323-ID = “cisco”)
Vendor Product ID Field: (productId = “ooh323”)

The below RPAD screenshot taken from an RPAD management interface shows both fields:



Solution

Firewalls:

Polycom cannot give advice on specific firewall configurations or specific firewalls. Those who have a third-party firewall as their defense against incoming H.323 calls should consult with their firewall and/or security experts before implementing any of the high-level recommendation in this advisory.

H.323 is not traditionally a firewall-friendly protocol. H.323 uses both UDP and TCP over a range of ports both static and dynamic. Firewalls that support H.323 natively include special rules or modules to deal with the nuances of the H.323 protocol. Such firewalls might be able to use the field information above in the form of rules or ACL's.

Alternatively, a firewall can come at the problem with more traditional IP-based whitelisting or blacklisting techniques.

RealPresence Access Director (RPAD):

RPAD administrators can make rules blocking for the two fields listed above. The rule for Product ID of “ooh323” is shown here.

Condition:

Attribute	Operator	Value
request.endpointVendor	==	ooh323

(request.endpointVendor.productId == "ooh323")

Detailed information about enabling and activating rules in RPAD is available in Appendix A at the end of this document.

Video Border Proxy (VBP):

VBP version 11.2.20 has fixed the issue without need for any configuration changes. This version disables fastStart on the WAN side when the call originates from a non-registered client.

This change appears to reduce the nuisance factor of the H.323 botnet without the need for configuration changes.

Revision History

DRAFT 1.0 – Original publication: November 6, 2014. RPAD configurations given with lighter reference to VBP.

Version 1.0 – Removed DRAFT status, cleared for public consumption, provided detail on VBP upgrade.

Version 1.1 – Clarification on advisory/bulletin differences and inclusion of *h323-ID = "cisco"* string earlier in the document.

©2014, Polycom, Inc. All rights reserved.

Trademarks

POLYCOM®, the Polycom logo and all names and marks associated with Polycom and Polycom's products are trademarks and/or service marks of Polycom, Inc. and are registered and/or common law marks in the United States and various other countries. All other trademarks are property of their respective owners. No portion hereof may be reproduced or transmitted in any form or by any means, for any purpose other than the recipient's personal use, without the express written permission of Polycom.

Disclaimer

While Polycom uses reasonable efforts to include accurate and up-to-date information in this document, Polycom makes no warranties or representations as to its accuracy. Polycom assumes no liability or responsibility for any typographical errors, out of



date information, or any errors or omissions in the content of this document. Polycom reserves the right to change or update this document at any time. Individuals are solely responsible for verifying that they have and are using the most recent Technical Bulletin.

Limitation of Liability

Polycom and/or its respective suppliers make no representations about the suitability of the information contained in this document for any purpose. Information is provided "as is" without warranty of any kind and is subject to change without notice. The entire risk arising out of its use remains with the recipient. In no event shall Polycom and/or its respective suppliers be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption, or loss of business information), even if Polycom has been advised of the possibility of such damages.

Appendix A: Detailed Instructions for Writing Rules in RPAD

Revealing the source H323-ID alias value of 'cisco': The RPAD Diagnostics->Call History->H323 Signaling type output shows them as:

Originator	Destination	Start Time	End Time	Signaling
H323_ID:cisco	DIALED_DIGITS:00220044122	2014-11-05 10:39:03 GMT-5	2014-11-05 10:39:03 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:00344122780	2014-11-05 10:31:02 GMT-5	2014-11-05 10:31:02 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:00644131507	2014-11-05 10:06:27 GMT-5	2014-11-05 10:06:27 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:00944122780	2014-11-05 09:35:03 GMT-5	2014-11-05 09:35:03 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:01144131507	2014-11-05 09:27:09 GMT-5	2014-11-05 09:27:09 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:01100441315	2014-11-05 09:05:29 GMT-5	2014-11-05 09:05:29 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:10441315070	2014-11-05 08:28:02 GMT-5	2014-11-05 08:28:02 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:10044131507	2014-11-05 08:20:24 GMT-5	2014-11-05 08:20:24 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:10004413150	2014-11-05 08:12:34 GMT-5	2014-11-05 08:12:34 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:10144122780	2014-11-05 08:04:32 GMT-5	2014-11-05 08:04:32 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:10114413150	2014-11-05 07:56:19 GMT-5	2014-11-05 07:56:19 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:11100441227	2014-11-05 07:48:20 GMT-5	2014-11-05 07:48:20 GMT-5	H.323
H323_ID:cisco	DIALED_DIGITS:12340044122	2014-11-05 07:24:37 GMT-5	2014-11-05 07:24:37 GMT-5	H.323

The screenshot shows the 'Details' view of a call. On the left, a sidebar lists 'Call Info', 'Call Events', and 'Subscription Events'. The main area displays 'Call Info' with the following details:

- Call status: Call Ended
- Start time: 2014-11-05 10:39:03 GMT-5
- End time: 2014-11-05 10:39:03 GMT-5
- Duration: 00:00:00
- Signaling: H.323

Below this, the 'Originator' and 'Destination' information is shown side-by-side:

Originator	Call ID: 9c5ef3c3-7dc4-1e40-09d1-c334aa5ffa88	Destination	Call ID: 9c5ef3c3-7dc4-1e40-09d1-c334aa5ffa88
	From: H323_ID:cisco		From: H323_ID:cisco
	To: DIALED_DIGITS:002200441227806181		To: DIALED_DIGITS:002200441227806181
	Dialed string: DIALED_DIGITS:002200441227806181		Dialed string: DIALED_DIGITS:002200441227806181
	IP address: 27.251.106.77		IP address: 27.251.106.77

Creating an RPAD ACL to prevent the unit from processing those inbound calls:

Configuration->Access Control List Rules

To create the new Rule

In the Actions pane click Add

The Add Rule dialog opens

Give the Rule a Name, such as MaliciousTraffic

Use the drop-arrow to select H323

Optionally, provide a Description such as “this Rule is intended to match on h323ID=cisco and will be used with a Deny action to block unwanted call attempts.”

In the Add Rule dialog, click Add to add a Condition

The Add Condition dialog opens

In the Add Condition dialog, use the drop-arrow for Attribute and select **request.srcAlias.firstH323-ID**

If the Operator is not == use the drop-arrow to select == (two equal symbols)

In the Value field enter “cisco” without the quotes.

Relation	Attribute	Operator	Value
	request.srcAlias.firstH323-ID	==	cisco

Click Ok to save the condition

The new Rule should appear similar to this:

Attribute	Operator	Value
request.srcAlias.firstH323-ID	==	cisco

(request.srcAlias.firstH323-ID == "cisco")

To apply the Rule, go to Configuration->Access Control List Settings

If there are no existing Settings, click Add in the Actions pane to open the Add ACL Setting dialog

For Service Name use the drop-arrow to select H323

For IP use the drop arrow to select the External Signaling IP of your RPAD

For Port use the drop-arrow to select H.225 call signaling port:1720

In the Rule Setting area click Add to open the Add Rule dialog

Use the drop-arrow under Access Control List Name to select the MaliciousTraffic Rule created in the previous step

Use the drop-arrow below Action to select Deny

Click Ok to add the Rule to the Rule Settings area

In the Rule Setting area, highlight the MaliciousTraffic Rule and then use the Priority Up button to move this rule to the top of the list; you may have to Priority Up more than once.

Click OK to save the Setting

The completed rule should appear similar to this example:

Service Name: H323

IP: External signaling IP : 140.242.225.1

Port: H.225 call signaling port : 1720

Rule Setting

Rule Name	Action
MaliciousTraffic	deny
H323_Guest_Call_Not_To_71xx	deny

Priority Down Priority Up

Add Edit Delete

OK Cancel Help

After applying the ACL to TCP1720, the next incident will to occur show a Call_End of TERMINATED_BY_RPAD.

RPAD Diagnostics->Call History->H323 Signaling type output

Click the call to select it, then click Show Call Details in the Actions pane

In the Call Details dialog, click Call Events to display the details

Event	Attributes	Time
CALL_BEGIN	Native-callid: 9c5ef3c3-7dc4-1e40-09d1-c334aa5ffa88 Source-uri: H323_ID:cisco Destination-uri: DIALED_DIGITS:002200441227806181 Signaling-type: H.323 Direction: INBOUND	2014-11-05 10:39:03.114 GMT-5
CALL_SIGNALING_EVENT	Event-type: INBOUND_REQUEST Far-end: 27.251.106.77:47944 Summary: setup Details: <input type="button" value="Show Message"/>	2014-11-05 10:39:03.115 GMT-5
CALL_END	Native-callid: 9c5ef3c3-7dc4-1e40-09d1-c334aa5ffa88 Reason: TERMINATED_BY_RPAD	2014-11-05 10:39:03.135 GMT-5
CALL_SIGNALING_EVENT	Event-type: OUTBOUND_REQUEST Far-end: 27.251.106.77:47944 Summary: releaseComplete Details: <input type="button" value="Show Message"/>	2014-11-05 10:39:03.136 GMT-5

4 records found

Note the sourceIP seen in the CALL_SIGNALING_EVENT for the Far-End

View the utility.log to verify the newly-created ACL is acting as desired.

Diagnostics->System Log files

Select the utility.log entry and click Download Logs to save the file to your computer

Diagnostics > System Log Files User name: LOCAL\admin Wednesday, November 05, 2014 11:06

Filter: Active logs

Time	Host	Filename	Size
2014-11-05 10:04	prod-rpad.polycomsupport.info	accessProxy.log	19244
2014-11-05 11:04	prod-rpad.polycomsupport.info	audit.log	307330
2014-11-05 10:05	prod-rpad.polycomsupport.info	dbAccess.log	472616
2014-11-05 11:02	prod-rpad.polycomsupport.info	h323Service.log	701943
2014-11-05 11:06	prod-rpad.polycomsupport.info	license.log	192331
2014-11-05 10:04	prod-rpad.polycomsupport.info	mediaTraversal.log	3554
2014-11-05 11:06	prod-rpad.polycomsupport.info	serviceController.log	1783137
2014-11-05 11:00	prod-rpad.polycomsupport.info	sipService.log	40359
2014-11-05 10:04	prod-rpad.polycomsupport.info	snmp.log	2068
2014-11-05 10:53	prod-rpad.polycomsupport.info	tunnel.log	15528
2014-11-05 11:00	prod-rpad.polycomsupport.info	utility.log	37802
2014-11-05 11:06	prod-rpad.polycomsupport.info	webAdmin.log	2375378

Use a text editor to search the utility.log for the sourceIP and you should see a line indicating the request was denied by the MaliciousTraffic ACL:

```

2 [20141105 06:58:16 664 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 27.251.150.44:40444 is denied by
3 [MaliciousTraffic]_H323_140.242.225.117_1720_1
4 [20141105 06:05:47 214 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 72.249.45.71:57828 is denied by
5 [MaliciousTraffic]_H323_140.242.225.117_1720_1
6 [20141105 06:13:20 933 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 83.96.159.206:34707 is denied by
7 [MaliciousTraffic]_H323_140.242.225.117_1720_1
8 [20141105 06:21:07 691 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - SIP request INVITE from 199.168.141.199:5060 is denied by
9 [MaliciousTraffic]_H323_140.242.225.117_1720_1
10 [20141105 06:23:35 913 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - SIP request SETUP from 199.27.89.22:35232 is denied by
11 [MaliciousTraffic]_H323_140.242.225.117_1720_1
12 [20141105 06:29:01 813 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - SIP request SETUP from 92.27.146.164:49836 is denied by
13 [MaliciousTraffic]_H323_140.242.225.117_1720_1
14 [20141105 06:37:02 849 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - SIP request INVITE from 62.210.78.210:5060 is denied by
15 [MaliciousTraffic]_H323_140.242.225.117_1720_1
16 [20141105 06:37:49 255 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 201.33.235.198:37853 is denied by
17 [MaliciousTraffic]_H323_140.242.225.117_1720_1
18 [20141105 06:45:06 855 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 212.182.57.150:53736 is denied by
19 [MaliciousTraffic]_H323_140.242.225.117_1720_1
20 [20141105 06:52:54 369 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 75.101.143.102:38146 is denied by
21 [MaliciousTraffic]_H323_140.242.225.117_1720_1
22 [20141105 07:08:53 988 INFO ] [GkEventExecutor:246] shared.accesscontroller.AccessController - H323 request SETUP from 27.251.106.77:50524 is denied by
23 [MaliciousTraffic]_H323_140.242.225.117_1720_1
24 [20141105 07:16:50 324 INFO ] [GkEventExecutor:300] shared.accesscontroller.AccessController - H323 request SETUP from 27.251.106.77:50524 is denied by
25 [MaliciousTraffic]_H323_140.242.225.117_1720_1
26 [20141105 07:24:37 045 INFO ] [GkEventExecutor:306] shared.accesscontroller.AccessController - H323 request SETUP from 77.68.36.13:47304 is denied by AC
27 [MaliciousTraffic]_H323_140.242.225.117_1720_1
    
```

Optionally, you can add a Condition to deny the traffic based on the product.VendorID, which has been seen as “ooh323”. To add this Condition, go to Configuration->Access Control List Rules, click your MaliciousTraffic Rule to select it, and click Edit in the Actions pane to open the Edit Rule dialog.

Using the steps details above, Add a Condition by first clicking the existing H323.request.dstAlias.firstH323-ID entry, then click the Add button to open the Add Condition dialog

Use the drop-arrow below Relation to select **or**

Use the drop-arrow below Attribute to select **request.endpointVendor.productId**



Use the drop-arrow below Operator to select == (two equal symbols)

In the Value field enter **ooh323**

This is two lower-case “o” characters, a lower-case “h” and the digits “323”

Click OK to add the new Condition

Click OK again to save the changes to the Rule; there is no need to re-apply the Rule in Access Control List Settings

