Asterisk Security

- Protect your investment

Asterisk Security

- Asterisk Configuration stupidity
- Asterisk hardening
- Privacy protection

Asterisk Configuration Stupidity

- Dial plan security
- SIP.conf
- IAX2.conf
- Manager.conf
- Billing problems

Dial plan security

- - Extension hopping
- - CallerID based protections
- - _.
- - Demo context
- - User access to the dial plan
- - Be careful with the default context
- - Limit simultaneous calls

Extension hopping

- User can reach ANY extension in the current context:

[internal]

exten => intro,1,Background(question);

exten => 1,spanish,Goto(Spanish)

exten => 2,english,Goto(English)

exten => _XX.,1,Dial(ZAP/g1/${EXTEN});

CallerID based protection

```
exten =>
    _X.,1,GotoIf($[“${CALLERIDNUM}”=“32134”?3);
exten => _X.,2,Hangup();
exten => _X.,3,Dial(${EXTEN});
```

- When not explicitly defined for each user/channel in zapata.conf, sip.conf, iax.conf, the user can choose his own CallerID!

call my way

Inappropriate use of _.

- _. Would match EVERYTHING!

(also fax, hang up, invalid, timeout,….)

Example:

exten => _.,1,Playback(blah);

exten => _.,2,Hangup;

➔ Causing a FAST LOOP.

(changed in CVS-head)

demo context

- Not a real security risk
- But… Someone might play with your system and use up your bandwidth, make prank calls to anyone…

User access to the dialplan

- - AMP and other GUI's might allow the ISP's user to change a dial plan in his own context. E.g.: hosted PBX's

- - Goto / GotoIf / dial(Local/…) -> context hopping.
- - System -> could do anything

Default context

- Example:

[default]
Include outgoing;
Include internal;

OH OH OH, guest calls will go to the default context!!!!!

Context usage:

- A call has two legs, the used context is the context defined for that user/channel in the config file for that protocol.

E.g:

- Zap to sip call:

  context set in zapata.conf is used

- SIP to IAX2 call:

  context in sip.conf is used

Context usage:

- In sip.conf, zapata.conf, iax2.conf…

A default context is defined, if there is no specific context setting for this channel or user, than the default context is used!

Limit simultaneous calls

- Sometimes you don’t want a user to make multiple simultaneous calls.

- E.g.: prepay / calling cards

Solution: setgroup, checkgroup (don’t trust incominglimit.)

exten => s,1,SetGroup(${CALLERIDNUM})

exten => s,2,CheckGroup(1)

Only good if the CallerID cannot be spoofed !!!!

Consider using accountcode for this.

Sip.conf

- Default context
- Bindport, bindhost, bindip
- [username] vs username=
- Permit, deny, mask
- Insecure=yes, very, no
- User vs peer vs friend
- Allowguest
- Autocreatepeer
- Pedantic
- Ospauth
- **Realm**
- Md5secret
- User authentication logic
- Username= vs [username]

Bindport, bindhost,bindip

- If you only use sip for internal calls, don’t put bindip=0.0.0.0 but limit it to the internal IP.

- Changing the bindport to a non 5060 port might save you from portscan sweeps for this port.

Permit, deny, mask

- Disallow everything, then allow per user the allowed hosts or ranges.

(Multiple are allowed.)

SIP.conf – insecure option

Insecure = …

- No: the default, always ask for authentication
- Yes: To match a peer based by IP address only and not peer.
- Insecure=very ; allows registered hosts to call without re-authenticating, by ip address
- Insecure=port; we don’t care if the portnumber is different than when they registered
- Insecure=invite; every invite is accepted.

User vs Peer vs Friend in SIP

- USER: never registers only makes calls
- PEER: can register + can make calls.

[user1]
type=user
[user1]
type=peer

Is allowed and the same as type=friend if the other parameters are identical!!!

Allowguest =…

- True: unauthenticated users will arrive in the default context as defined in sip.conf
- False: unauthenticated users will get a permission denied error message.
- OSP: to allow guest access for voip traffic coming from an OSP server.

autocreatepeer

- The autocreatepeer option allows, if set to Yes, any SIP UA to register with your Asterisk PBX as a peer. This peer's settings will be based on global options. The peer's name will be based on the user part of the Contact: header field's URL.

  This is of course a very high security risk if you haven't got control of access to your server.

Pedantic

- Defaults to pedantic=no
- If enabled, this might allow a denial of service by sending a lot of invites, causing a lot of (slow) DNS lookups.

Realm

- Realm=Asterisk; Realm for digest authentication

; Defaults to “Asterisk"

; Realms MUST be globally unique according to RFC 3261

; Set this to your host name or domain name

How is authentication done?

- chan_sip.c: /* Whoever came up with the authentication section of SIP can suck my %*!#$ for not putting an example in the spec of just what it is you're doing a hash on. */

How is authentication done?

- Look at FROM header in SIP message for the username:

-> browse sip.conf for a type=user with that username
If found -> check the md5
If not found,

-> browse sip.conf for a type=peer with that username
-> browse sip.conf for an (registered) IP where the request is coming from
if insecure=very, no more checks are done
if insecure=port, if they are willing to authenticate, even if they are calling from a different port than they registered with. (used for NAT not using the same port number every time).
otherwise, check the md5 + allow/deny.

- If no peer found ? do we allow guest access (allowguest=true ?)
- Yes? OK, allow send it to the default context, if not reject.

Secret vs md5secret

- With SIP all passwords are md5 encrypted when sending the packets, but are stored in plaintext in sip.conf

- [user]
- Secret=blabla

Secret vs md5secret

- echo - n "<user>:<realm>:<secret>" | md5sum
- E.g.:

echo -n "user:asterisk:blabla" | md5sum

e1b588233e4bc8645cc0da24d8cb848d

[user]

md5secret=e1b588233e4bc8645cc0da24d8cb848d

Username= vs [username]

- [username] is for authentication a client connecting to asterisk.

Username=… is to have your asterisk server authenticate to another SIP server.

Iax.conf

- auth=plaintext,md5,rsa
- User authentication logic
- Default context
- [username] vs username=
- Permit, deny, mask
- Bindport, bindhost, bindip
- User vs peer vs friend

iax.conf - auth

- Plaintext: passes are sent in plaintext
- Md5: encrypt the password with md5
- RSA: use public key / private key – uses AES.

User vs Peer vs friend

- USER: can only accept calls
- PEER: can only make calls
- FRIEND: can do both

[user1]
type=user
[user1]
type=peer

Is allowed!!!

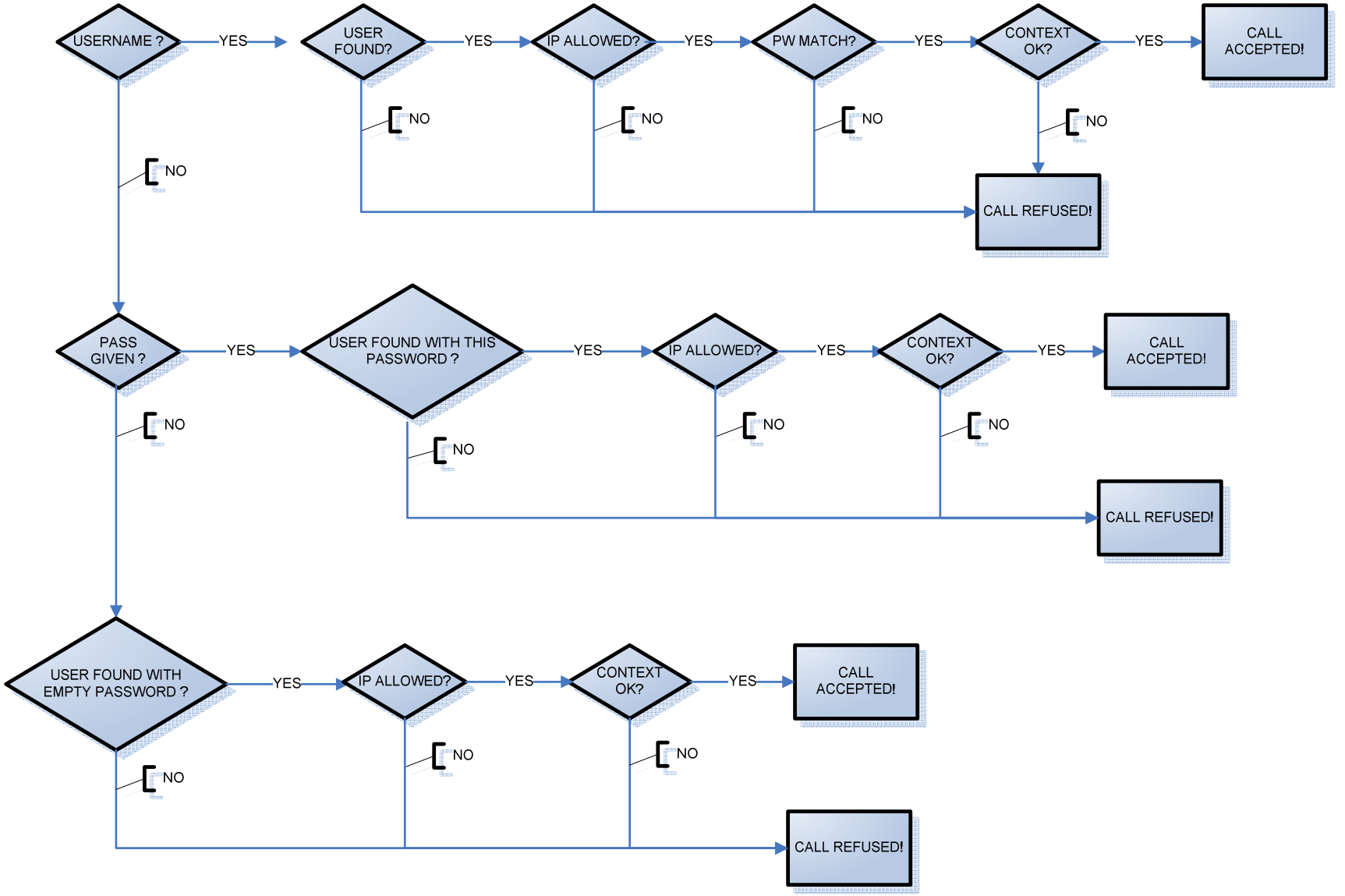
How is authentication done?

- In iax2: (cvs-head!!)

Pseudocode:
Is username supplied ?
-> yes -> matched against iax.conf users starting bottom to top.
user found ?
-> yes : is IP in allowed / disallowed list ?
yes –> does password match ?
yes -> does requested context match a context=… line?

-> no -> is a password given ?
-> yes : Asterisk will look bottom to top for a user with this password,
-> if the context matches, or there is no context specified, and the host is in the allowed lists (allow / deny) then the call is accepted.

-> no: Asterisk will look bottom to top for a user without password.
-> if the context matches, or there is no context specified, and the host is in the allowed lists (allow / deny) then the call is accepted.



- Add a last entry in iax.conf with no password to force nosecret access into a specific context.
- If you use realtime, don’t have any user without a password and without permit/deny.

Manager.conf

```
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0

[zoa]
secret = blabla
deny=0.0.0.0/0.0.0.
permit=221.17.246.77/255.255.255.0
permit=127.0.0.1/255.255.255.0
read = system,call,log,verbose,command,agent,user
write = system,call,log,verbose,command,agent,user
```

Manager.conf

- No encryption is used, even the password is sent in plaintext.

- Don’t enable it on a public IP.
- Use http://www.stunnel.org/
- Watch out with management programs with direct interface to the manager.
- Limit the privileges per user (especially the system!!!).

Asterisk Security

- Asterisk Configuration stupidity
- Asterisk hardening
- Privacy protection

Asterisk Hardening

- Asterisk as non-root user
- Asterisk in CHROOT
- Asterisk in a JAIL
- Asterisk with limited read / write permissions
- ZAPTEL kernel modules
- Asterisk firewalling / shaping / NAT
- Tty9
- Linux hardening
- Remote logging
- Tripwire
- Limit running system processes

call my way

Asterisk as non root user

adduser --system --home /var/lib/asterisk --no-create-home Asterisk
chown -r asterisk:asterisk /var/lib/asterisk
chown -r asterisk:asterisk /var/log/asterisk
chown -r asterisk:asterisk /var/run/asterisk
chown -r asterisk:asterisk /var/spool/asterisk
chown -r asterisk:asterisk /dev/zap
chown -r root:asterisk /etc/asterisk

chmod -r u=rwX,g=rX,o= /var/lib/asterisk
chmod -r u=rwX,g=rX,o= /var/log/asterisk
chmod -r u=rwX,g=rX,o= /var/run/asterisk
chmod -r u=rwX,g=rX,o= /var/spool/asterisk
chmod -r u=rwX,g=rX,o= /dev/zap
chmod -r u=rwX,g=rX,o= /etc/asterisk
chown asterisk /dev/tty9
su asterisk -c /usr/sbin/safe_asterisk
Or Asterisk -U asterisk -G asterisk

Asterisk with limited read / write permissions

- Asterisk has no write permissions for its config files and is running as non root ?

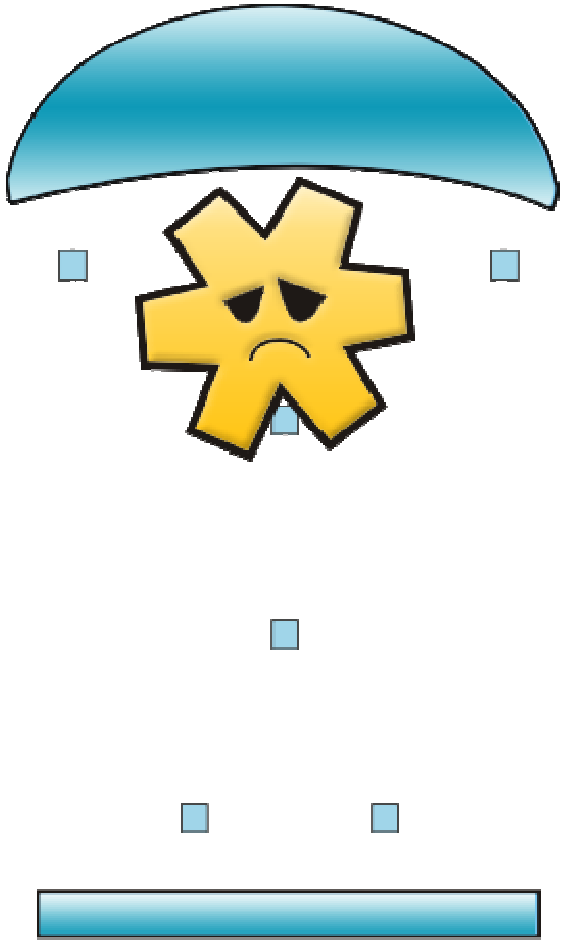
- In the unlikely event of someone breaking in through Asterisk, your dial plan is still vulnerable through the CLI or the manager.

Asterisk in chroot

- Changes the root directory visible to asterisk to e.g. /foo/bar
- Pretty useless if asterisk is running as root and perl or gcc is available.

Asterisk in a jail

- Changes the root directory visible to Asterisk.
- Limits the commands / programs any user in this jail can execute to a list you specify.
- Expansion of chroot.



Zaptel kernel modules

- Zaptel is module only, cannot be put into the kernel.
- Hackers like to hide in a module, they can backdoor a module, compile it, load it in memory and remove all traces on the disk.
- You could have the kernel check an md5 for the Zaptel modules.
- I think Matt Frederickson compiled them in the kernel before.

Firewalling / shaping / NAT

- Block everything except the ports you really want. (5060, 4569, …)
- RTP ports are a big pita (see rtp.conf)

Sidenote: you might want to check your ISP is not blocking anything in the range defined in RTP.conf

Limit access to tty9

- safe_asterisk opens a console on tty9.

This does not require a password and will provide a root shell to anyone passing by.

(by using !command on the CLI).

- Remove the offending line, or don’t use safe_asterisk

Linux Hardening

- GRsec (2.6.x)
- Openwall (2.4.x)
- Remove all unneeded things.

Remote logging

- Remote syslog
- Put Asterisk log files (and other log files on a remote server).

Tripwire

- Make hashes of all the important files on the server and check them for changes you didn’t do.

Limit server processes

- An Asterisk server should be only:

- OS + ASTERISK.
- No database
- No APACHE
- No PHP

(If you really need those, and don’t have enough servers, don’t put them on a public IP and firewall them!!!!)

Asterisk Security

- Asterisk Configuration stupidity
- Asterisk hardening
- Privacy protection

Asterisk privacy

- Encryption
- Monitoring
- CallerID spoofing
- CallingPRES

Call Encryption - SIP

- SRTP -> method to encrypt voice packets.
- TLS -> method to encrypt signaling packets.

➔Both are not yet supported by asterisk.

➔Bounty on voip-info.org.

Call Encryption – IAX2

- 30/12/2004 2:07

Modified Files: chan_iax2.c iax2-parser.c iax2-parser.h iax2.h Log Message: Minor IAX2 fixes, add incomplete-but-very-basically-functional IAX2 encryption.

It would support any type of encryption you like.
-> Doesn’t work yet.

Call Encryption – General solution

- Send you packets through a VPN or tunnel.

- Use only UDP tunnels to avoid delays.

Known to work:

➔IPSEC, VTUN, OPENVPN.

Call Encryption – Tunnel solution

➔Advantage, CPU expensive encryption can happen on dedicated machine.

➔ Disadvantage: doesn't work on hardphones or ATA's without adding an extra server in front of them.