RTC Security

Jason Ostrom, Security Consultant

Stora

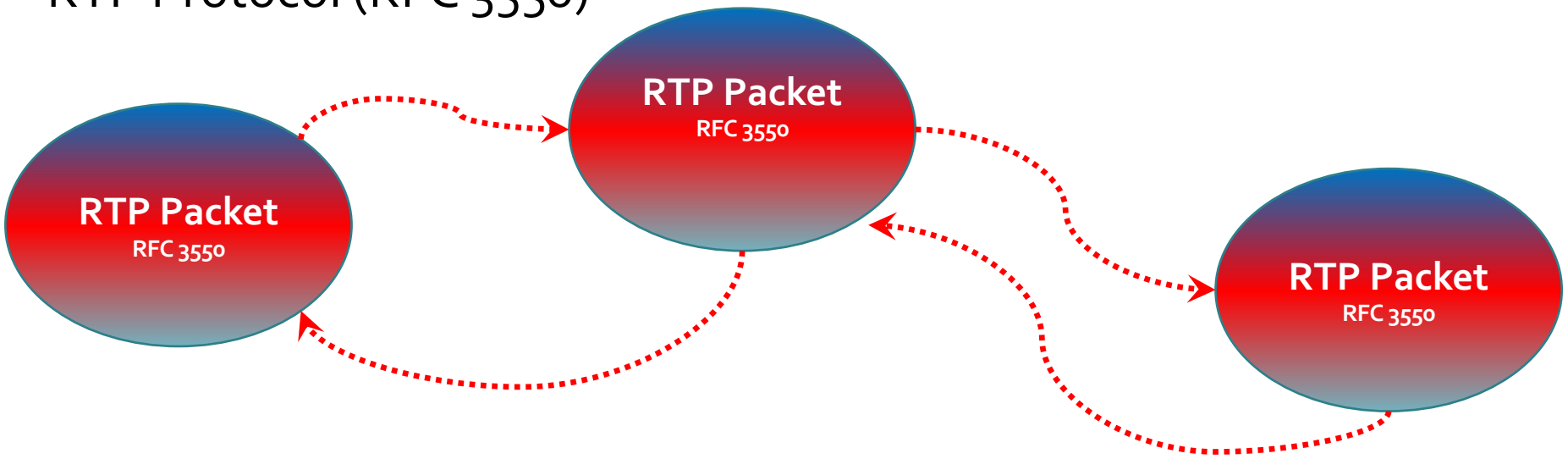
About Me

- Assessments
- Security Research
- Coder
- Stora

```
root@kali-sans:~# voiphopper -V
VoIP Hopper 2.04
Copyright (C) 2012 Jason Ostrom  <jpo@pobox.com>
Location:  http://voiphopper.sourceforge.net
root@kali-sans:~#
```

What is RTC?

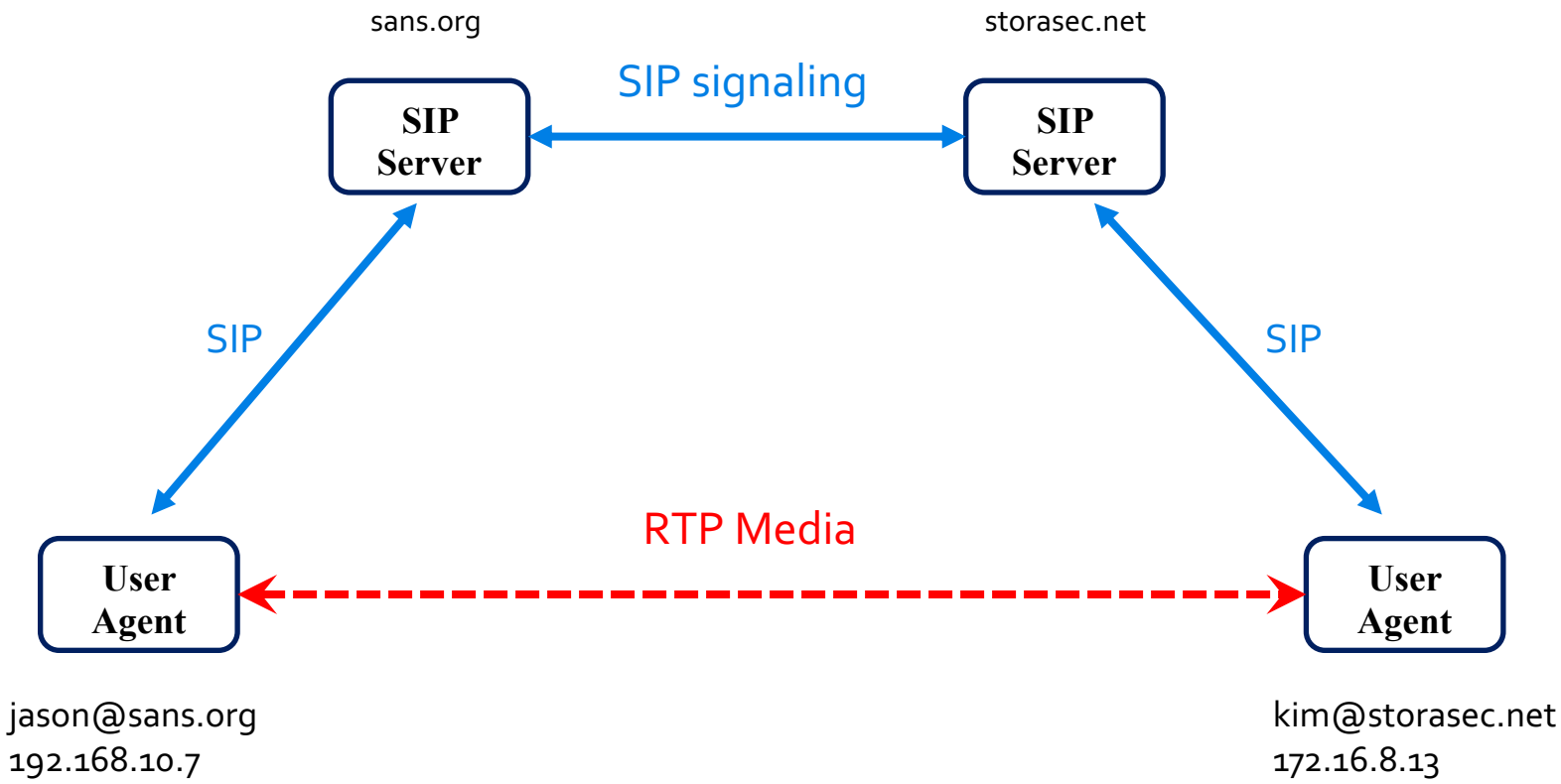
- Real-time communications
  - Any application with a real-time requirement for media transfer
  - RTP Protocol (RFC 3550)



Agenda

- ✓SIP Security
  - ✓Intro to SIP
  - ✓SIP Trunk Architectures 101
  - ✓Security Risks and Attacks
- WebRTC

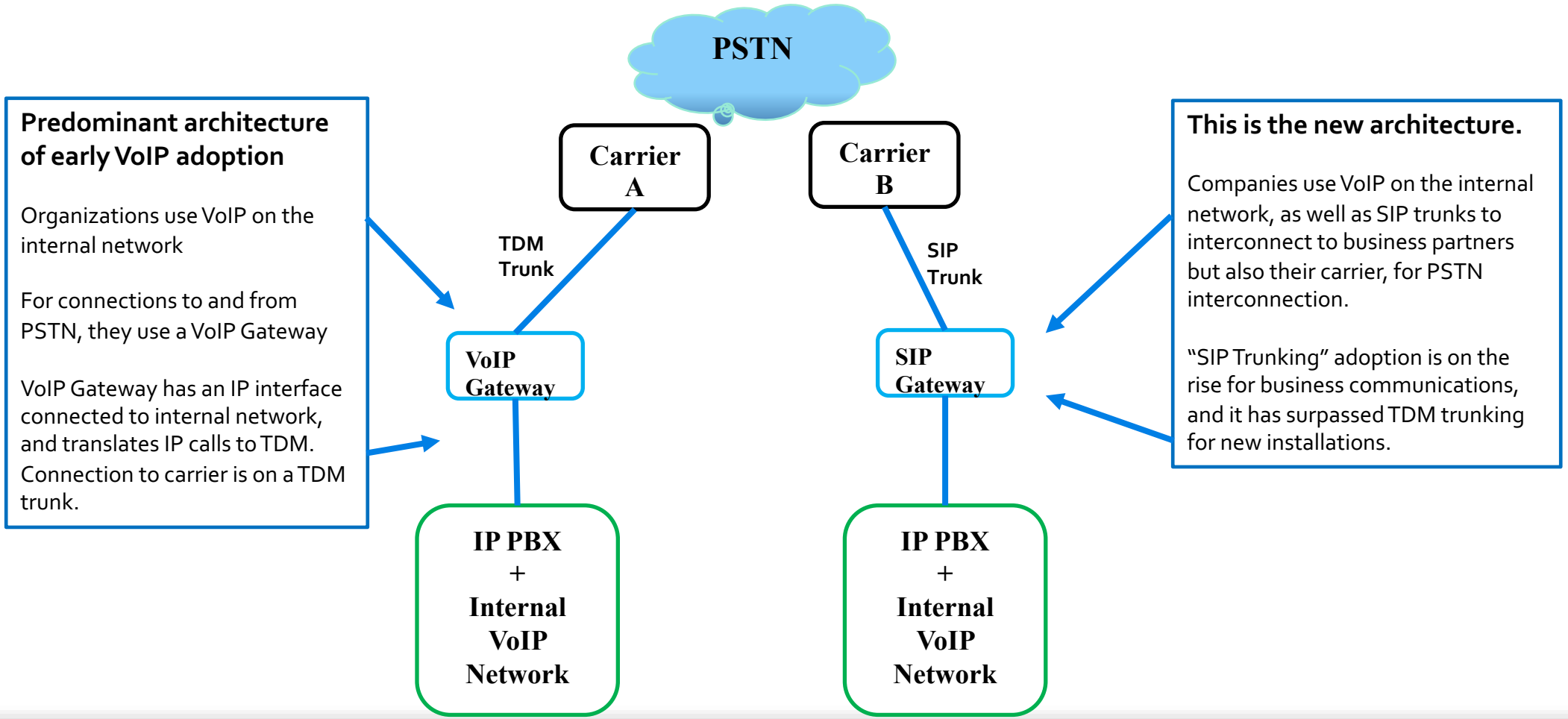
Intro to SIP: SIP Basics



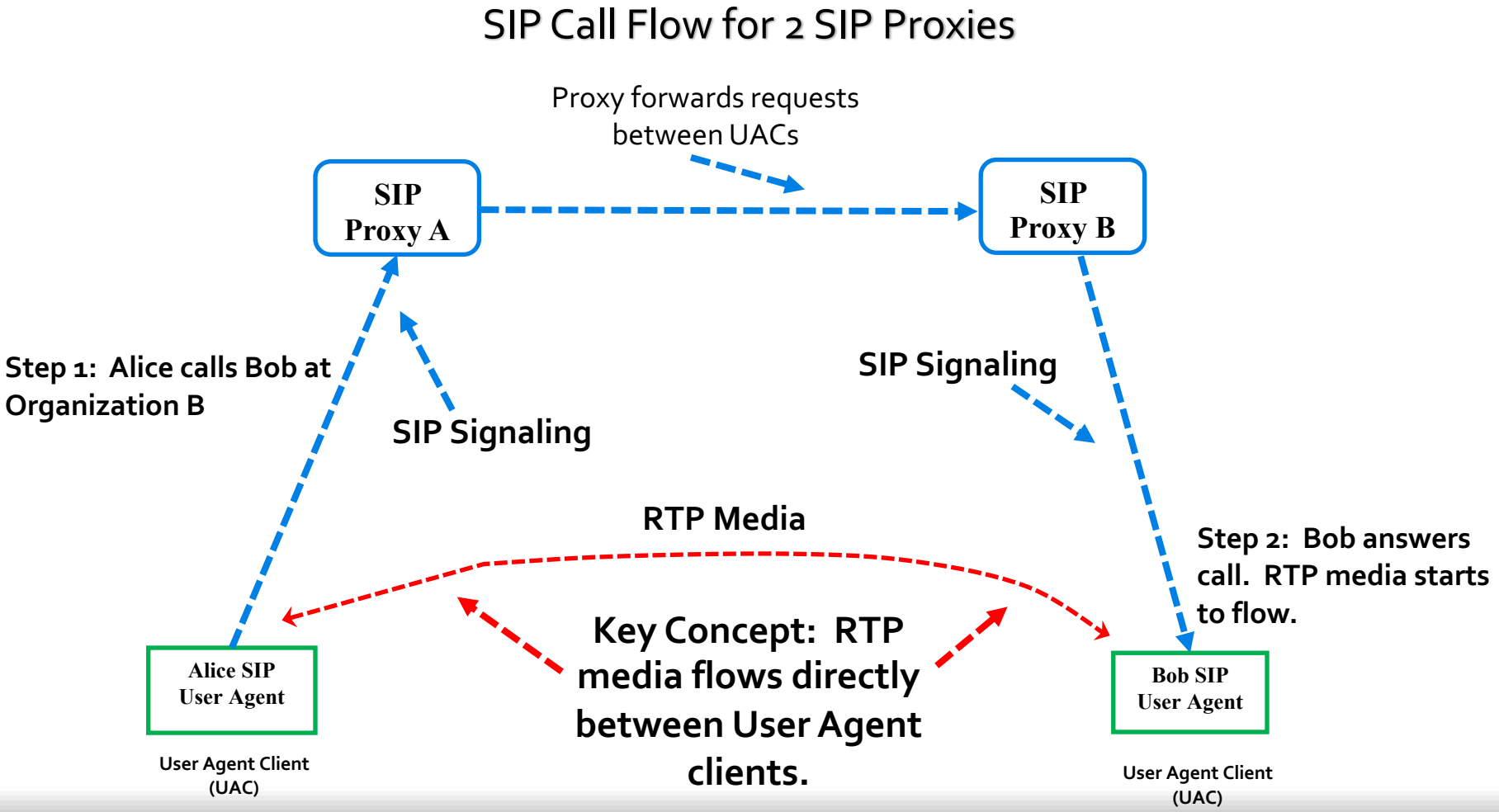
SIP Trunking (1)

- SIP Trunking
  - Objective: A "SIP Trunk" is a growing service and business application; important to understand architecture basics, to understand risk exposure areas
  - Definition: A SIP trunk is a service which allows a SIP peering connection between an organization and a carrier, usually to allow PSTN connectivity
    - Several business benefits, including reduced telecom costs and toll bypass
    - Increasingly adopted by businesses, replacing TDM trunking

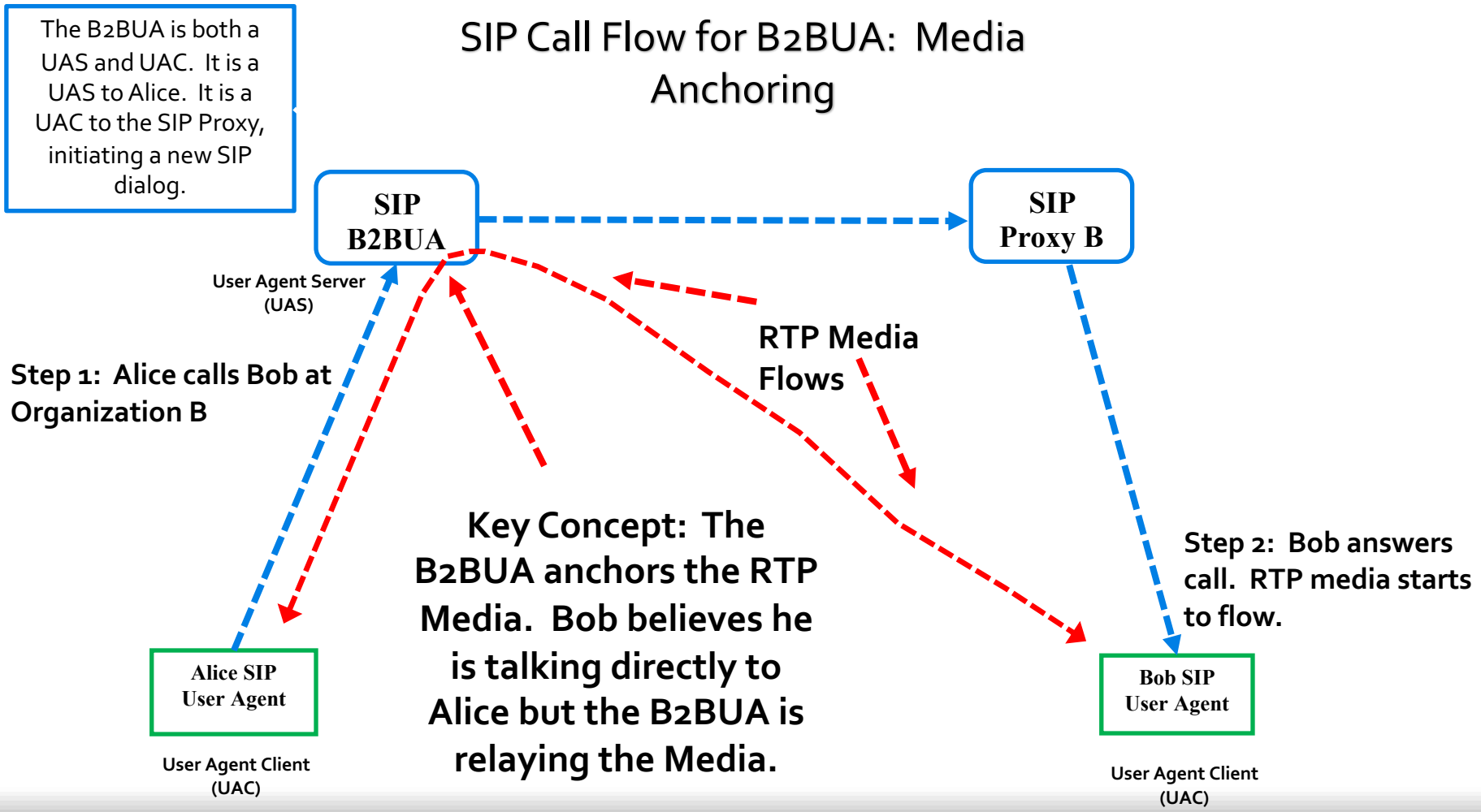
SIP Trunking (2)



Call Flow: Proxies



Call Flow: B2BUA



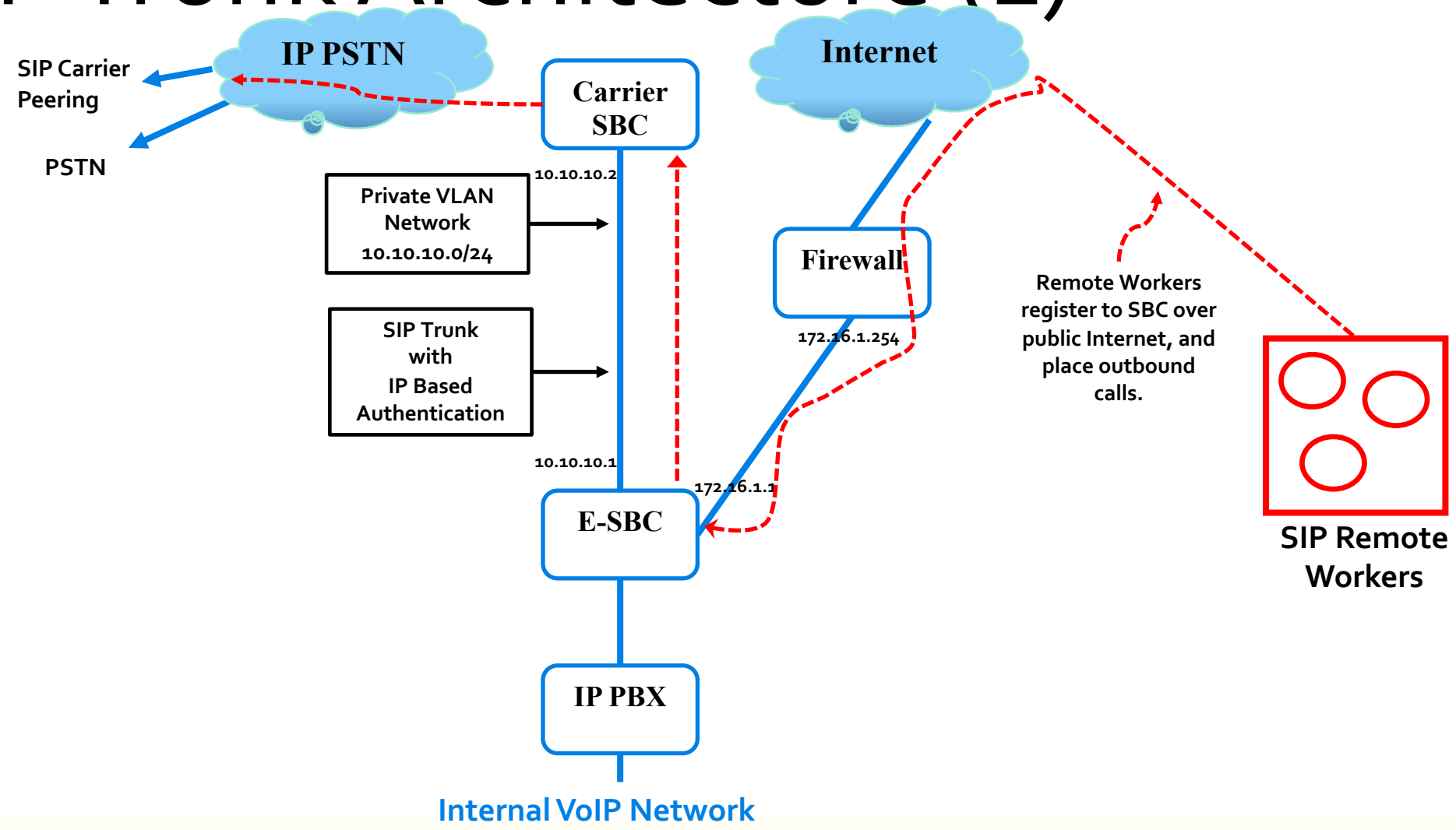
SBC: Session Border Controller

- Session Border Controller (SBC)
  - SBC is a market term. It is a commercial implementation of a B2BUA.
  - SBC is a popular SIP network element that is used as a border Gateway demarcation between SIP administrative domains
  - E-SBC: Enterprise SBCs. Used by enterprises to control their traffic.
- Provides important features and functionality
  - NAT Traversal (Far End NAT Traversal, Near End NAT Traversal) for RTP media pinhole behind a NAT, Topology Hiding
  - Provides Proxy, Registrar, and B2BUA services in single box
  - Security
  - Codec transcoding
  - Media Anchoring and control of Routing and Media

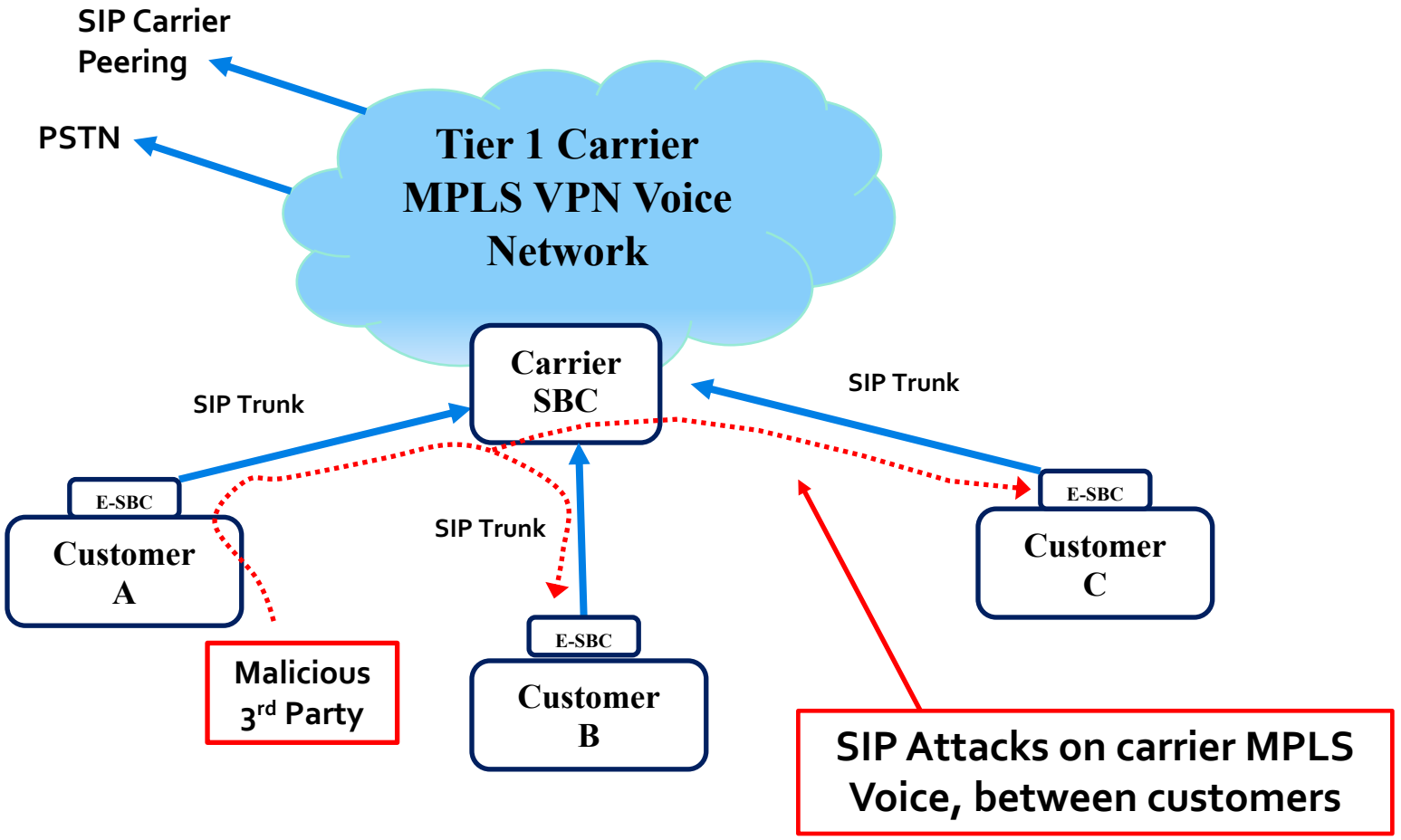
2 Types of SIP Trunk Authentication

- IP Based Authentication
  - Carrier only allows SIP connections from Source IP address of customer equipment
  - Usually used on private IP Network connections between an enterprise CPE and Carrier SBC
  - Usually used when a SIP trunking customer has purchased Network/Data transport and SIP trunking from same provider/carrier
  - RFC 1918 Private IP, or public IP addresses that are not routed on carrier network
  - Also referred to as "using ACLs" to only allow trusted SIP peers
  - Customer normally uses INVITE without authentication to send calls
- MD5 Digest Authentication
  - SIP carrier provides SIP credentials for authentication
  - Can be used from any IP address on public Internet
  - Also referred to as allowing "dynamic registrations"
  - Customer normally provides authenticated REGISTER, and each INVITE is authenticated

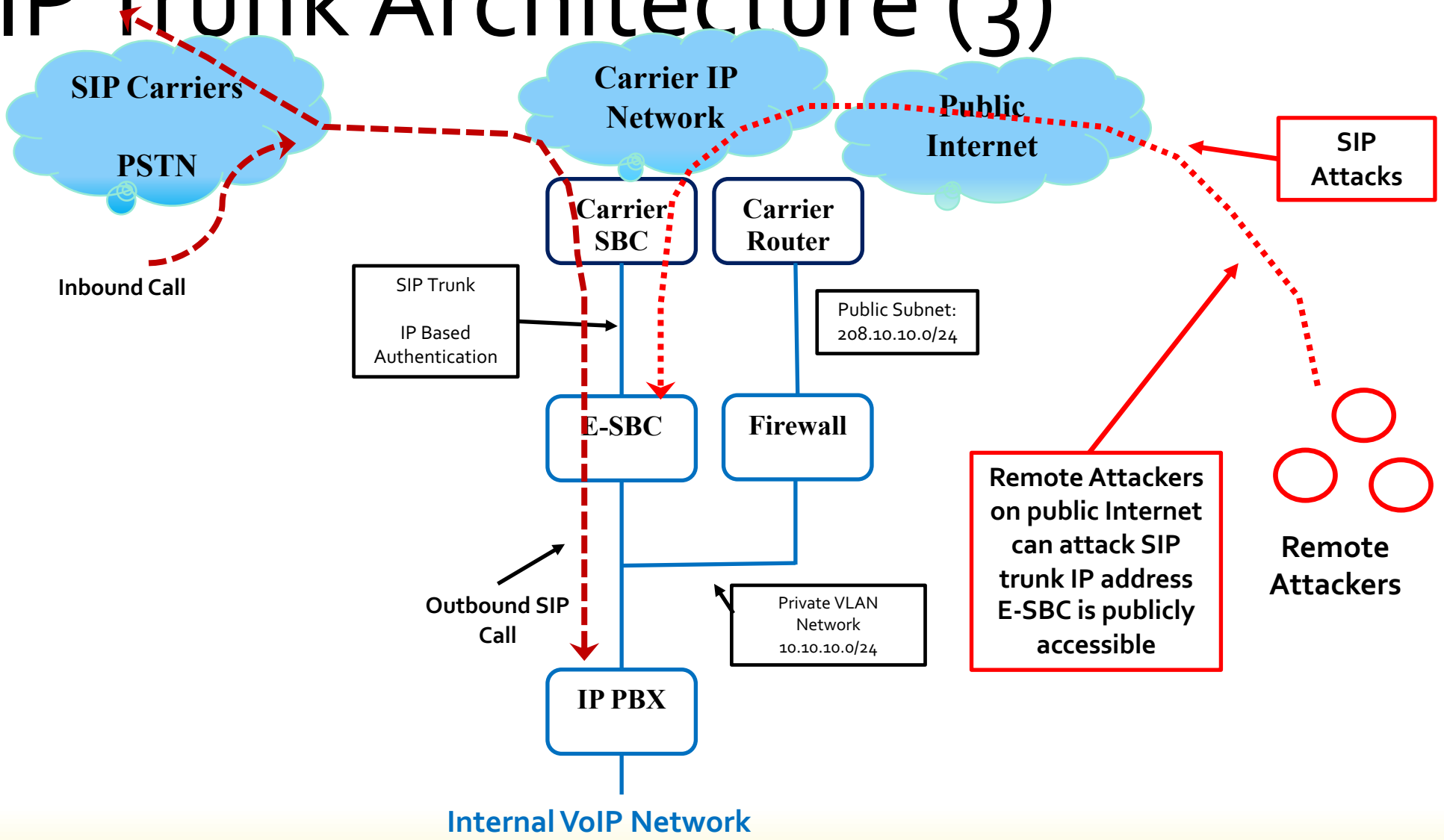
SIP Trunk Architecture (1)



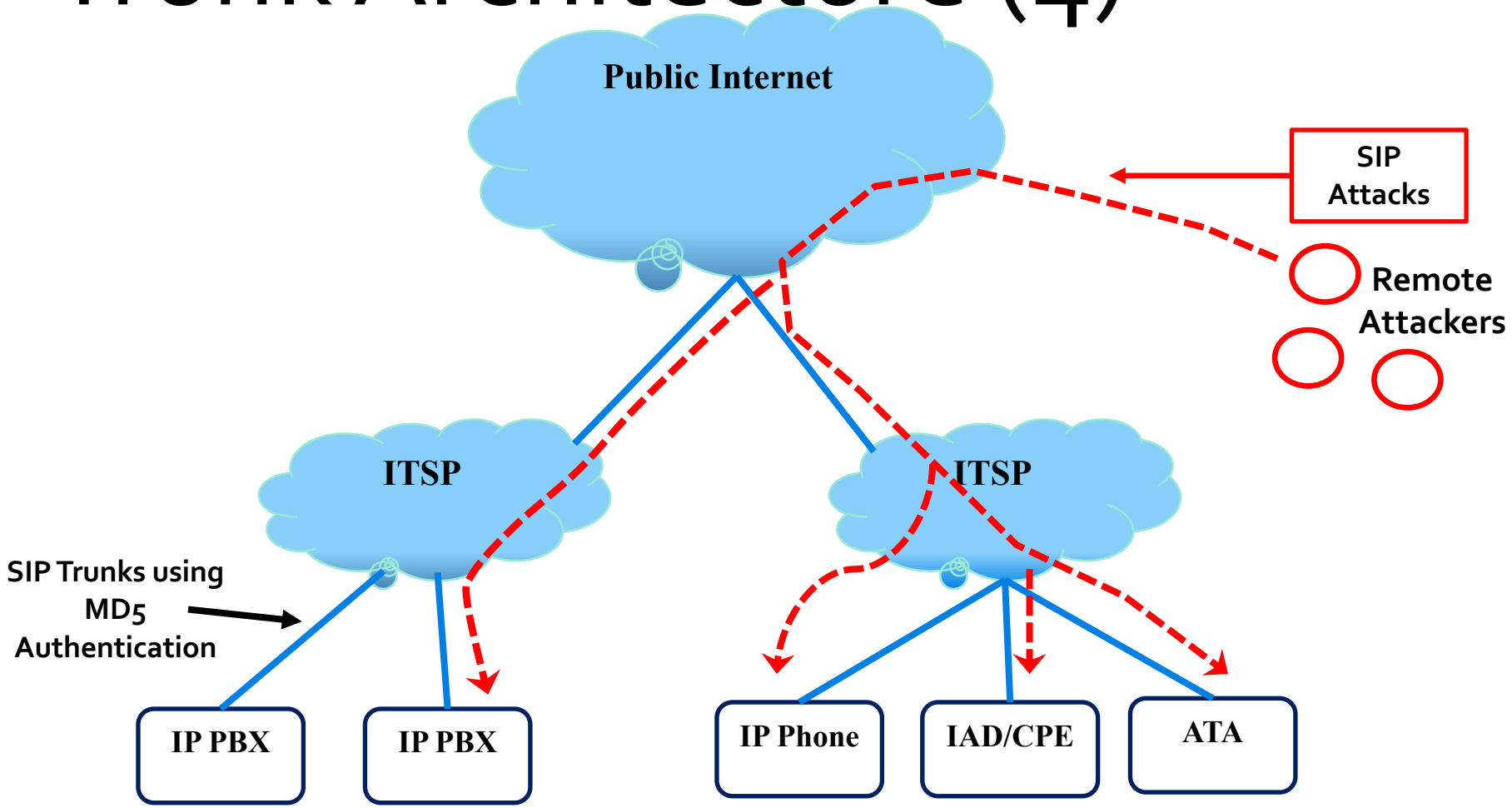
SIP Trunk Architecture (2)



SIP Trunk Architecture (3)



SIP Trunk Architecture (4)



Security Risks & Attacks

- Pentesting SIP Trunks
- Basic Methodology for assessing VoIP Networks
- Step 1: Mapping the VoIP Application
  - Understand protocols in use
  - Understand business logic, users
  - Understand VoIP application flow and network ingress/egress points
  - Understand SIP Trunking and where it is used
- Step 2: Threat Modeling for VoIP specific Attacks
  - Impact: DoS
  - Impact: Fraud

Cleartext SIP & MD5 Authentication

- With SIP Trunking:
  - Usage of unencrypted SIP can lead to increased risk when weak SIP passwords are used with MD5 Digest Authentication
  - Check to ensure if SIP trunk calls can be sniffed with access to the network

SIP uses MD5 digest authentication, which is vulnerable to an offline wordlist attack, if captured.

The digest response is shown to the right in the Authorization header

SIP Authorization Header

```
Authorization: Digest username="4000", realm="50.116.8.147", nonce="Uyh911MofKuf
    Authentication Scheme: Digest
    Username: "4000"
    Realm: "50.116.8.147"
    Nonce Value: "Uyh911MofKufQkddARvUB1hJ6XnMoD/f"
    Authentication URI: "sip:50.116.8.147"
    Digest Authentication Response: "3a82fb6492a96c8cc32a3423256ed769"
    Algorithm: MD5
```

Verify: Vulnerable to offline MD5 wordlist Attack (1)

- SIPCrack tool suite
  - Suite of tools written by Martin J. Muench to dump SIP logins and crack credentials
  - Step 1: Use "sipdump" tool to dump the md5 hash from the authentication sequence, taking a pcap file or stdin as input
  - Usage: sipdump –p <file> <dump file>

```
root@sans:/opt/offline-md5auth-crack# sipdump -p sip-md5-auth-register1.pcap auth-output

SIPdump 0.2  ( MaJoMu | www.codito.de )
-------------------------------------

* Using pcap file 'sip-md5-auth-register1.pcap' for sniffing
* Starting to sniff with packet filter 'tcp or udp'

* Dumped login from [redacted] -> 172.16.200.2 (User: '3002')

* Exiting, sniffed 1 logins
root@sans:/opt/offline-md5auth-crack#
```

Verify: Vulnerable to offline MD5 wordlist Attack (2)

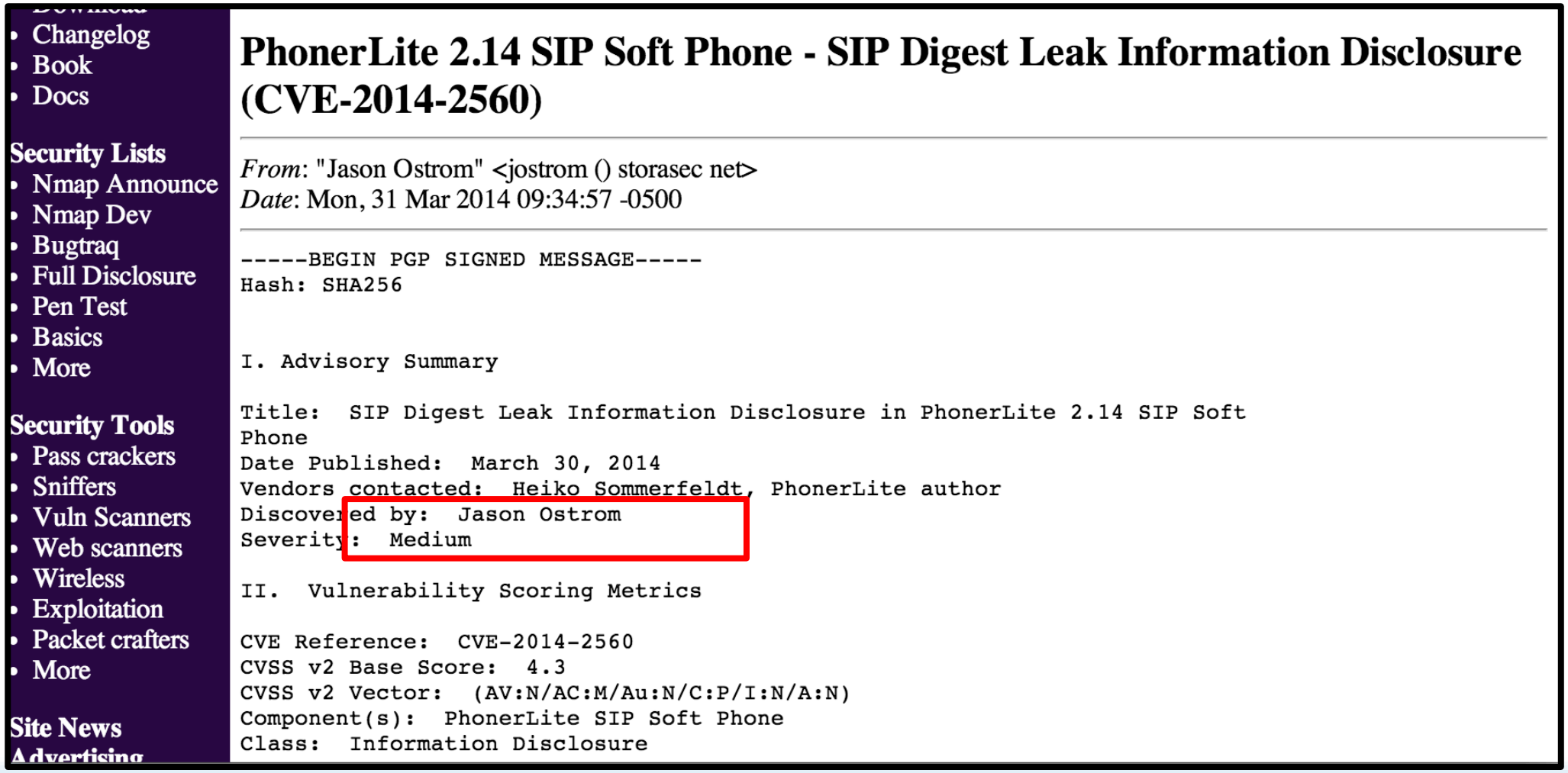
- Step 2: Use "sipcrack" tool to mount wordlist attack against dumped md5 hash
  - Usage: sipcrack –w <wordlist> <dump file>

```
root@sans:/opt/offline-md5auth-crack# sipcrack -w english.txt auth-output

SIPcrack 0.2  ( MaJoMu | www.codito.de )
---------------------------------------

* Found Accounts:

Num     Server          Client          User    Hash|Password

1       172.16.200.2    [redacted]      3002    3ab4a41abcb9408954e69bff86249960

* Select which entry to crack (1 - 1): 1

* Generating static MD5 hash... 2861599dbf2b004c4c3dadc5199f8e54
* Loaded wordlist: 'english.txt'
* Starting bruteforce against user '3002' (MD5: '3ab4a41abcb9408954e69bff86249960')
* Tried 45143 passwords in 0 seconds

* Found password: '18000'
* Updating dump file 'auth-output'... done
root@sans:/opt/offline-md5auth-crack#
```

SIP Digest Leak Information Disclosure

- Some SIP implementations are vulnerable to a MD5 digest leak
  - One can crack the password when they receive the password in an MD5 hash
  - This vulnerability was originally discovered by Sandro Gauci as a protocol implementation flaw, with soft phones and handsets showing different results in protection
    - https://resources.enablesecurity.com/resources/sipdigestleak-tut.pdf
  - Steps:
    - Malicious 3rd party attacker sends an INVITE, as a UAC, towards victim
    - If callee victim answers, wait until they hang up (like a dead air call)
    - When the SIP User Agent Server (UAS) sends the BYE, attackers send a challenge of 407 Proxy Authorization Required, or a 401 Unauthorized challenge
    - If vulnerable, SIP User Agent will reply with hashed MD5 password, in 2nd BYE
    - Dumping the hashed MD5 password from BYE, we can then mount wordlist attack

Vulnerability in SIP soft phone



- Changelog
- Book
- Docs

Security Lists
- Nmap Announce
- Nmap Dev
- Bugtraq
- Full Disclosure
- Pen Test
- Basics
- More

Security Tools
- Pass crackers
- Sniffers
- Vuln Scanners
- Web scanners
- Wireless
- Exploitation
- Packet crafters
- More

Site News

PhonerLite 2.14 SIP Soft Phone - SIP Digest Leak Information Disclosure (CVE-2014-2560)

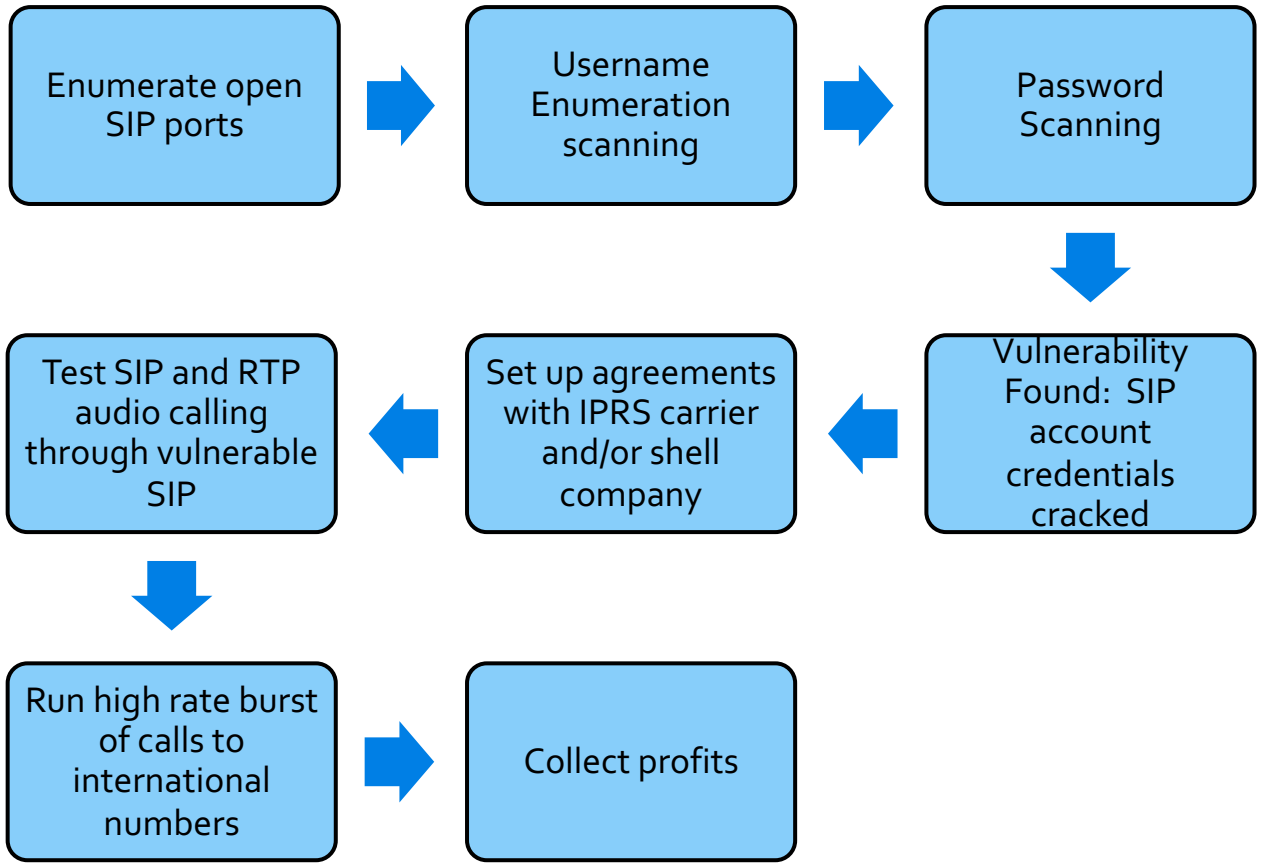
From: "Jason Ostrom" <jostrom () storasec net>
Date: Mon, 31 Mar 2014 09:34:57 -0500

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256


I. Advisory Summary

Title:  SIP Digest Leak Information Disclosure in PhonerLite 2.14 SIP Soft
Phone
Date Published:  March 30, 2014
Vendors contacted:  Heiko Sommerfeldt, PhonerLite author
Discovered by:  Jason Ostrom
Severity:  Medium

II.  Vulnerability Scoring Metrics

CVE Reference:  CVE-2014-2560
CVSS v2 Base Score:  4.3
CVSS v2 Vector:  (AV:N/AC:M/Au:N/C:P/I:N/A:N)
Component(s):  PhonerLite SIP Soft Phone
Class:  Information Disclosure
```

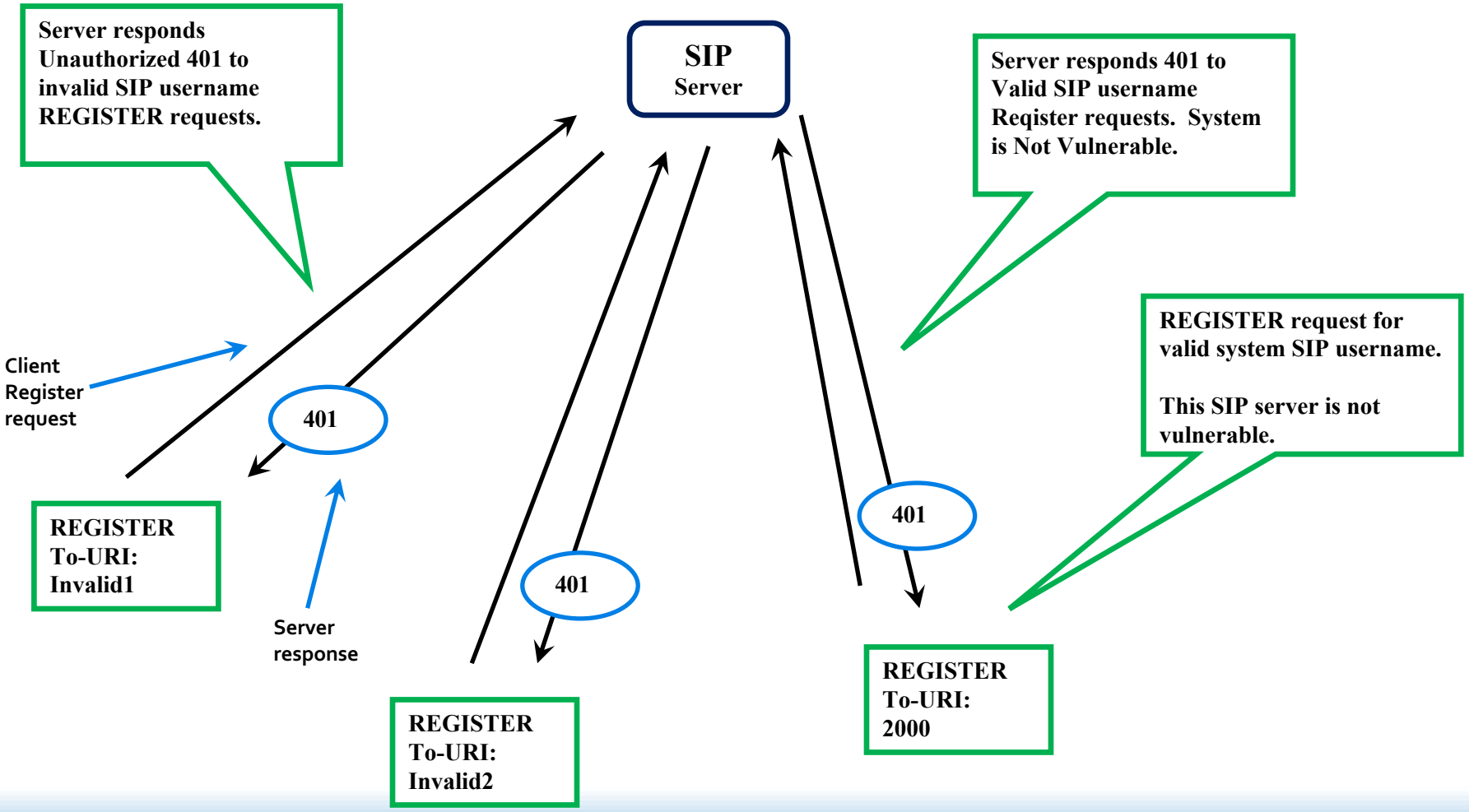
Cracking weak SIP credentials using scanning techniques

- One can exploit weak SIP usernames and passwords to carry out Fraud, over the remote public Internet
- Toll Fraud using International Premium Rate Services (IPRS) + open SIP
  - Greatest financial threat to companies using SIP on public Internet
  - This form of fraud is highly profitable to malicious 3rd parties
  - How this works
    - Malicious 3rd parties sets up shell company
    - Buys special premium rate phone numbers through a carrier
    - Finds vulnerable SIP servers on public Internet, testing calls
    - Runs call through vulnerable SIP servers to premium rate numbers

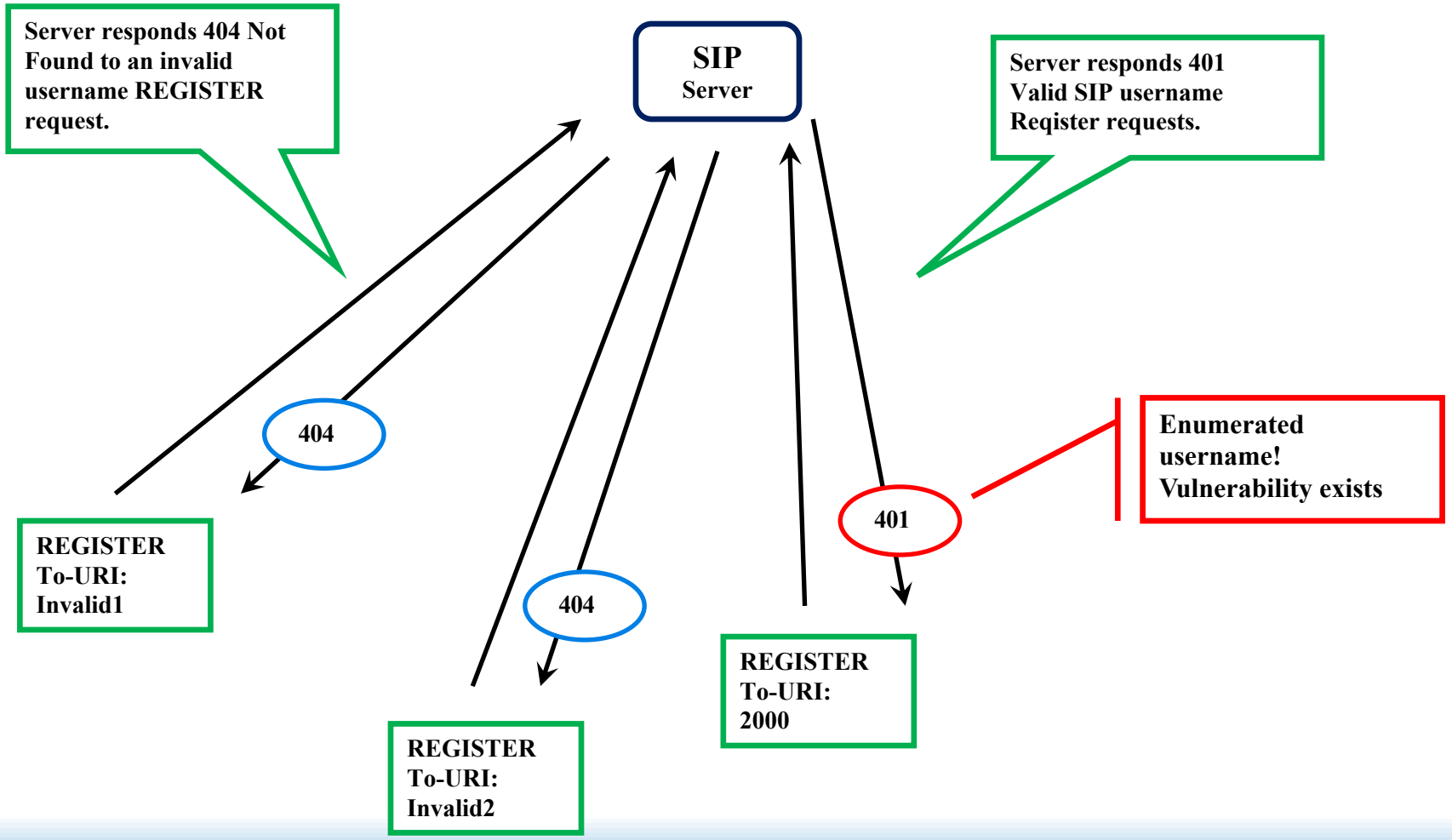
Toll Fraud Attacks for PSTN Dialing



SIP Username Enumeration(1)



SIP Username Enumeration(2)



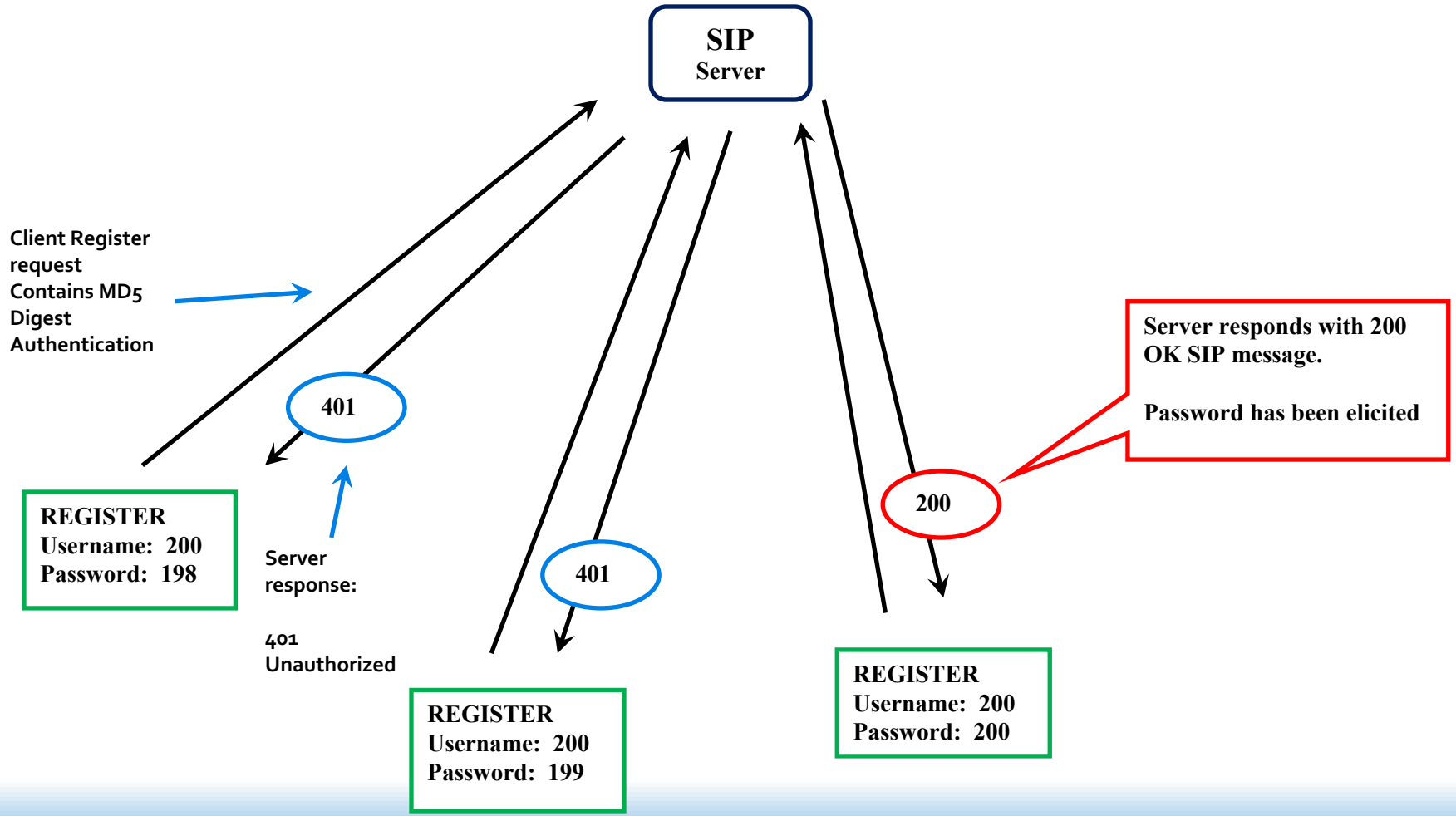
SIP Username Enumeration (3)

- After you have discovered the SIP server you wish to target, we need to harvest a list of SIP usernames, for further SIP attacks
- Tool Method: SIPVicious svwar.py tool
  - SVWar is included by default on Kali Linux
  - Download: https://code.google.com/p/sipvicious
- Usage
  - #svwar 192.168.10.1 –e100-2000
- With a list of valid SIP usernames, ready for next phase in attack methodology

SIP Credential Cracking (1)

- Exploiting a vulnerability to find a valid password, given a known SIP username
  - Vulnerability: SIP Password = Weak. Brute force guessing the password based on an input dictionary, over network services. Easily brute forced.
    - Password based on username (same as username, partial, reversed)
    - Password based on dictionary word, easily scanned for, guessed, vendor default
    - Password based on easily incremented numeric digits
  - Vulnerability: Security control failure in detecting, logging, blocking password scanning attacks
  - Scanning attacks very prevalent on public listening SIP services

SIP Credential Cracking (2)



SIP Credential Cracking (3)

- Given a known SIP username, we attempt to elicit the password
- Tool Method: SIPVicious svcrack.py tool
  - SVCrack is included by default on Kali Linux
  - Download: https://code.google.com/p/sipvicious
- Usage
  - #svcrack –u100 –d dictionary.txt 192.168.10.1
- If the SIP password can be cracked, it enables authenticated user SIP attacks

IP Based Authentication Spoofing Vulnerability

- SIP Trunks using IP Based Authentication
  - Can be vulnerable to IP address spoofing under certain conditions
    - UDP Transport
      - UDP is the most common transport for SIP trunking
    - The IP address is accessible over public Internet
  - Spoofing IP address on SIP trunks is trivial and results in Fraud, in two scenarios
    - Spoof carrier SBC IP address, for inbound calling
      - Results in IP Phones on internal network ringing
    - Spoof E-SBC IP address, send call to carrier SBC, for outbound calling
      - Call will ring to PSTN

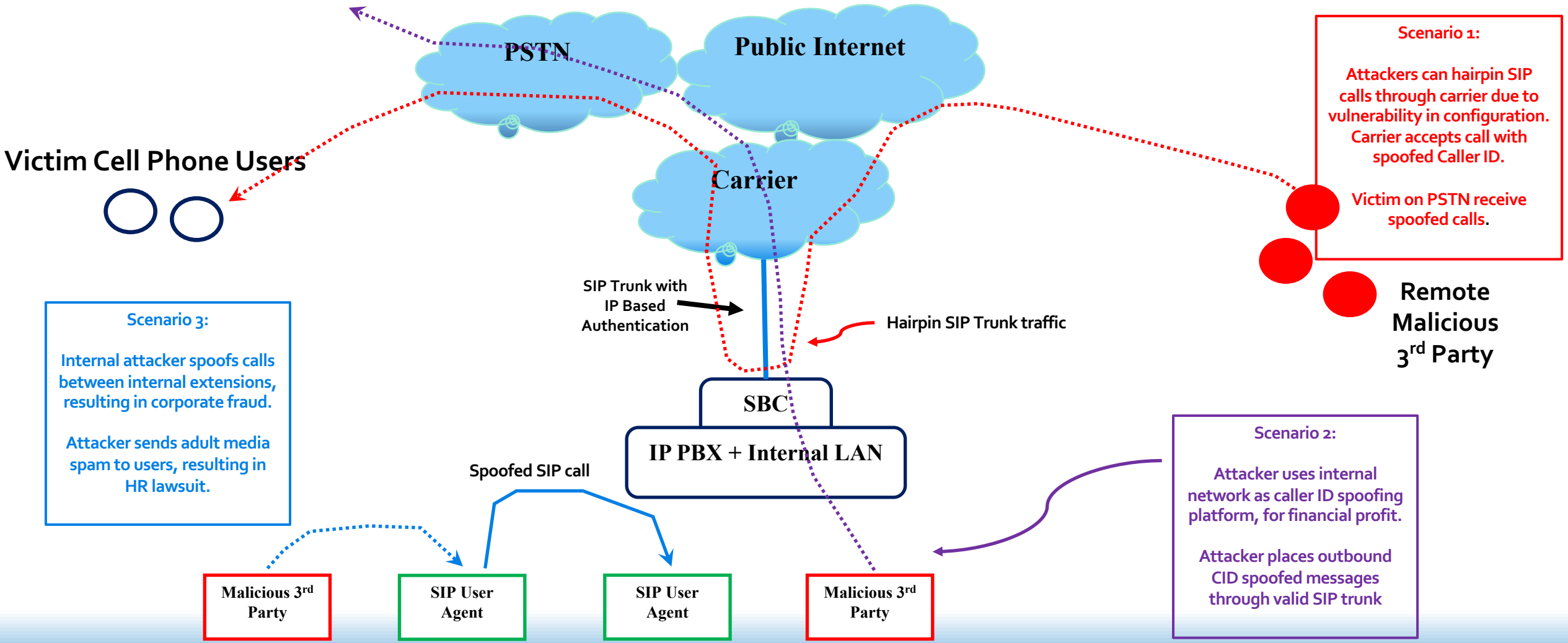
Carrier Fraud with IP Based Auth

- Attack vector: "Hairpin SIP Traffic"
  - SIP Trunk IP Based Authentication has transitive trust, which can be exploited by re-directing a SIP call
  - This form of attack is actively exploited: Mis-configured SIP CPE devices
    - When the carrier allows IP Based authentication from a customer's CPE
    - CPE has public IP address
    - Remote attacker simply spoofs a SIP INVITE and sends it to CPE
    - If CPE is mis-configured to accept SIP peering from any IP address, it hairpins INVITE call to carrier. Source IP of INVITE from CPE
    - From carrier perspective, the call looks like it originates from the customer premise.
    - Carrier can't easily distinguish call spoofed from Internet and hairpinned from legitimate customer originating traffic

SIP Caller ID Spoofing

- Vulnerability exploited: When a malicious 3rd party can spoof a parameter in the SIP header that shows a different numeric telephone number to callee
  - It's important to be able to verify and/or demonstrate this vulnerability against SIP trunks
    - Outbound dialing with caller ID spoof to PSTN, through SIP trunk
    - Bouncing a CID spoofed call over public Internet, through vulnerable SIP trunk
    - CID spoofing on internal network, between valid users, as SIP line side attack
  - Business Impact: Fraud, Corporate Espionage, DoS
  - Real World Attack Scenarios:
    - After you have cracked the SIP credentials for a valid user, you can do malicious calling of other users, spoofing their From Username
    - Enhanced social engineering

Threat Modeling: Caller ID Spoofing



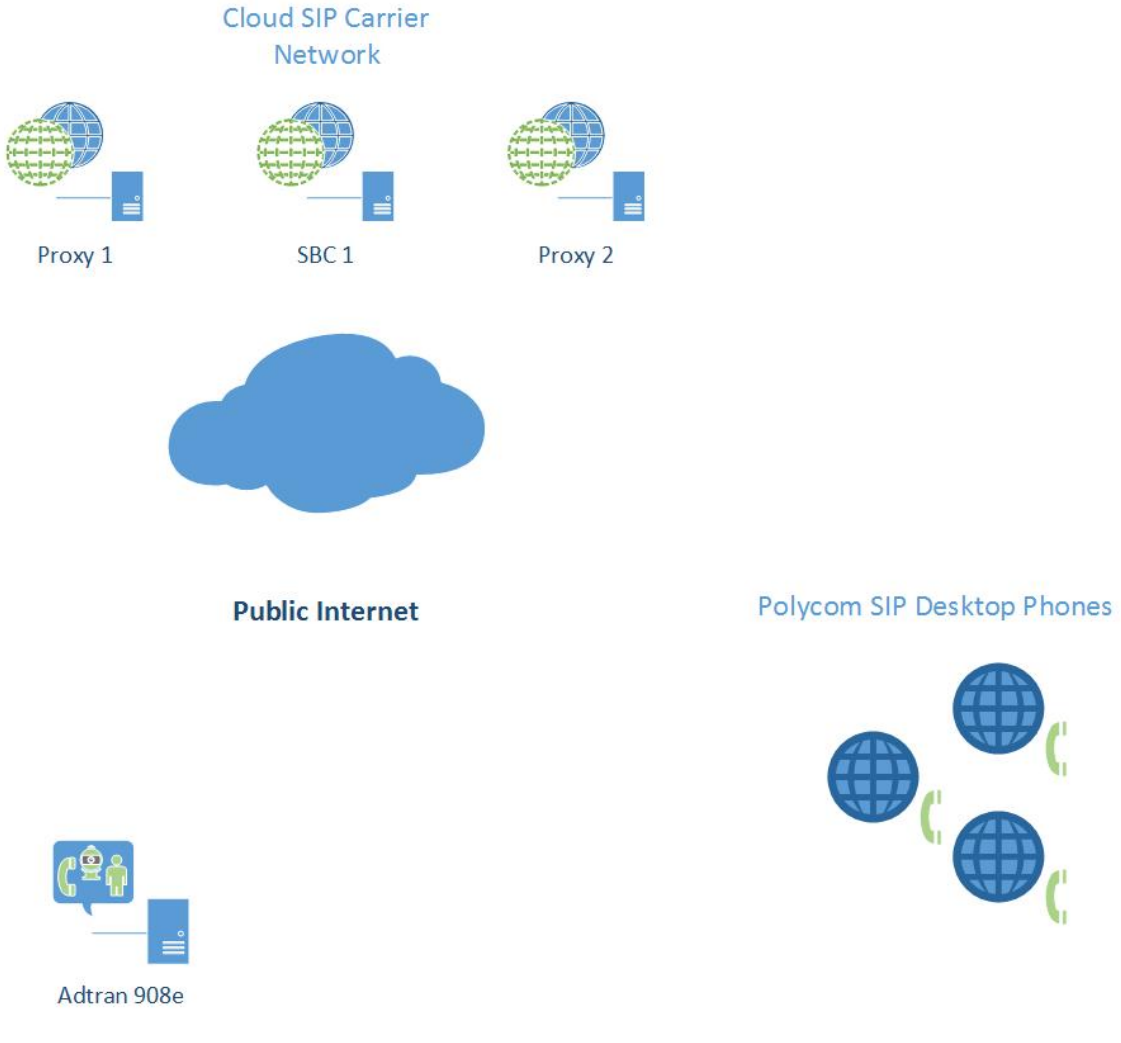
Takeaways (1)

- SIP Trunk Security Risks
  - SIP Protocol Implementation (SIP or SIP TLS + SRTP?)
    - SIP vulnerable to md5 offline brute force; Can capture authentication md5 hash?
    - Are weak SIP passwords in use? Can be cracked with wordlist?
    - Vulnerable to known SIP protocol implementation flaws (CVE-ID)?
  - SIP Trunk Authentication
    - With MD5 Auth:
      - Scanning techniques to divulge usernames?
      - Do SIP gateways block username/scanning attacks?
    - With IP Based Auth:
      - Vulnerable to spoofing attacks, such as hairpin of SIP traffic
      - Vulnerable to IP Based Spoofing, when UDP transport is used
  - Caller ID Spoofing
    - Spoofing From-URI Header, Remote-Party-ID, or spoofing with IP Based Auth

Takeaways (2)

- Security Best Practices / Mitigations
  - Always prefer enabling SIP TLS on SIP Trunks, if business can allow it
  - Ensure that SIP protocol implementation and vendor is not vulnerable to any known protocol flaws, based on CVE-ID
  - If MD5 Authentication is used on SIP Trunks, ensure that strong SIP password policy is enforced
    - Including policies for creation of SIP passwords, provisioning protocols
  - If IP Based Authentication is used + UDP Transport
    - Consider enabling MD5 + IP Based Auth
    - You can never be too safe, because UDP SIP messages can always be spoofed
  - Test IP Based Authentication SIP trunks for hair pin message spoofing attacks

RTC Security Lab Network
rtcsecurity.net



Live security tools demo

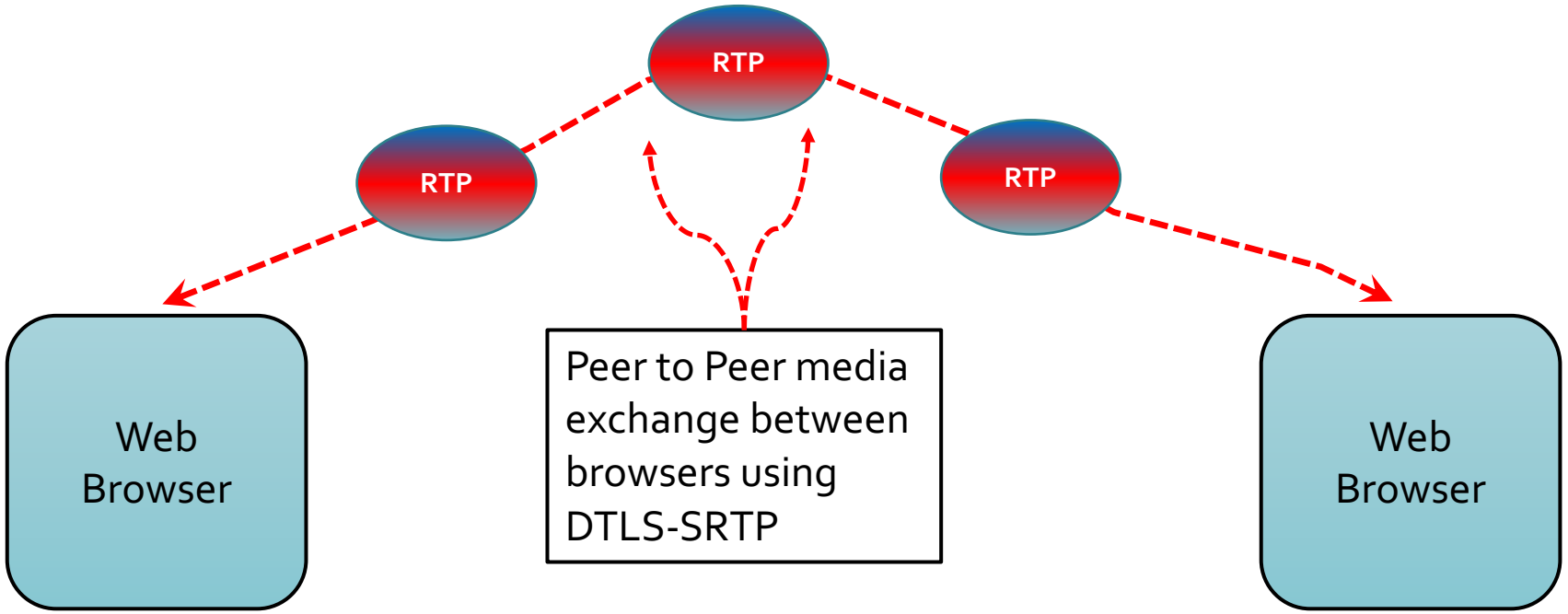
Agenda

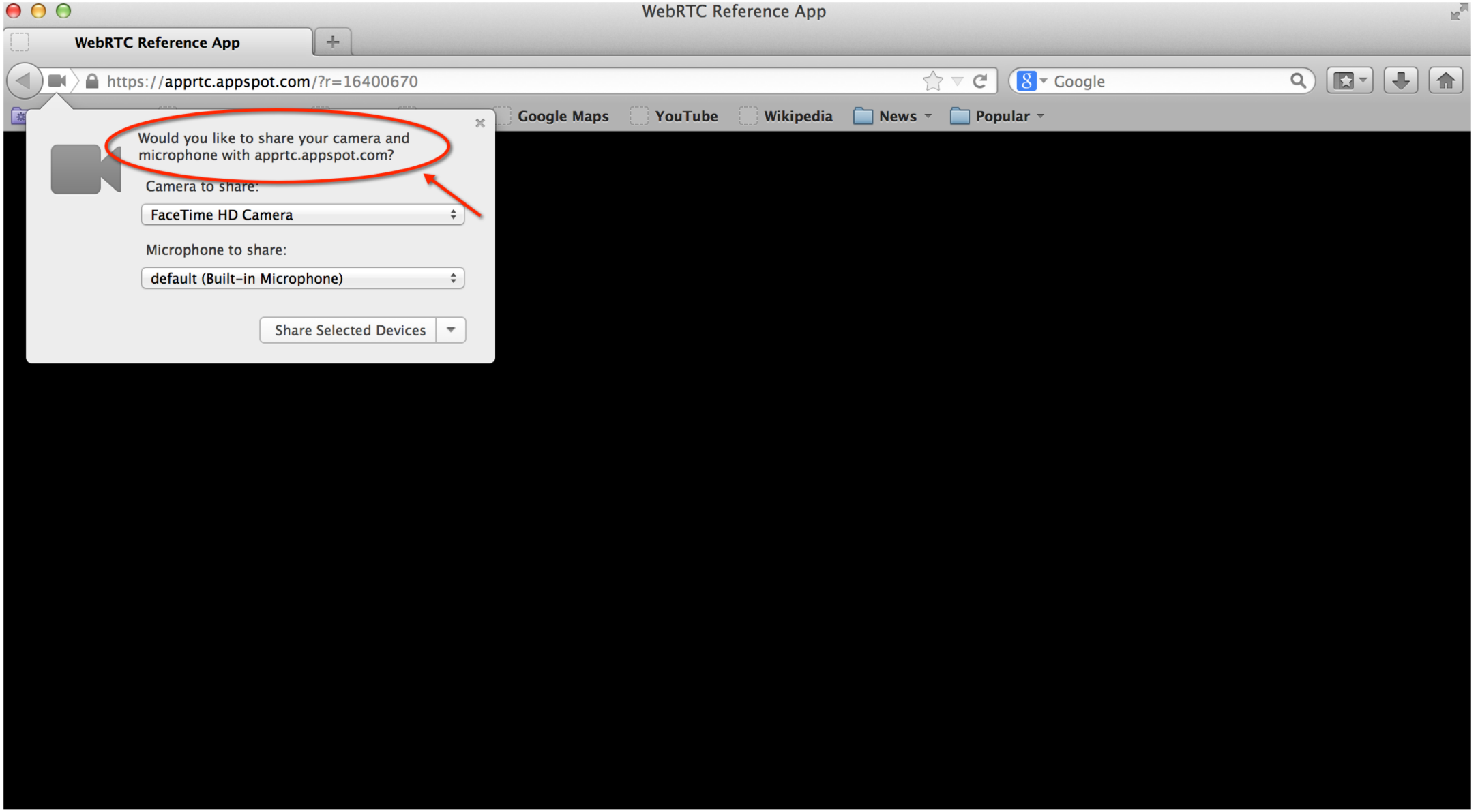
- SIP Security
- ✓WebRTC
  - ✓Intro
  - ✓Architecture
  - ✓Security Framework

What is WebRTC?

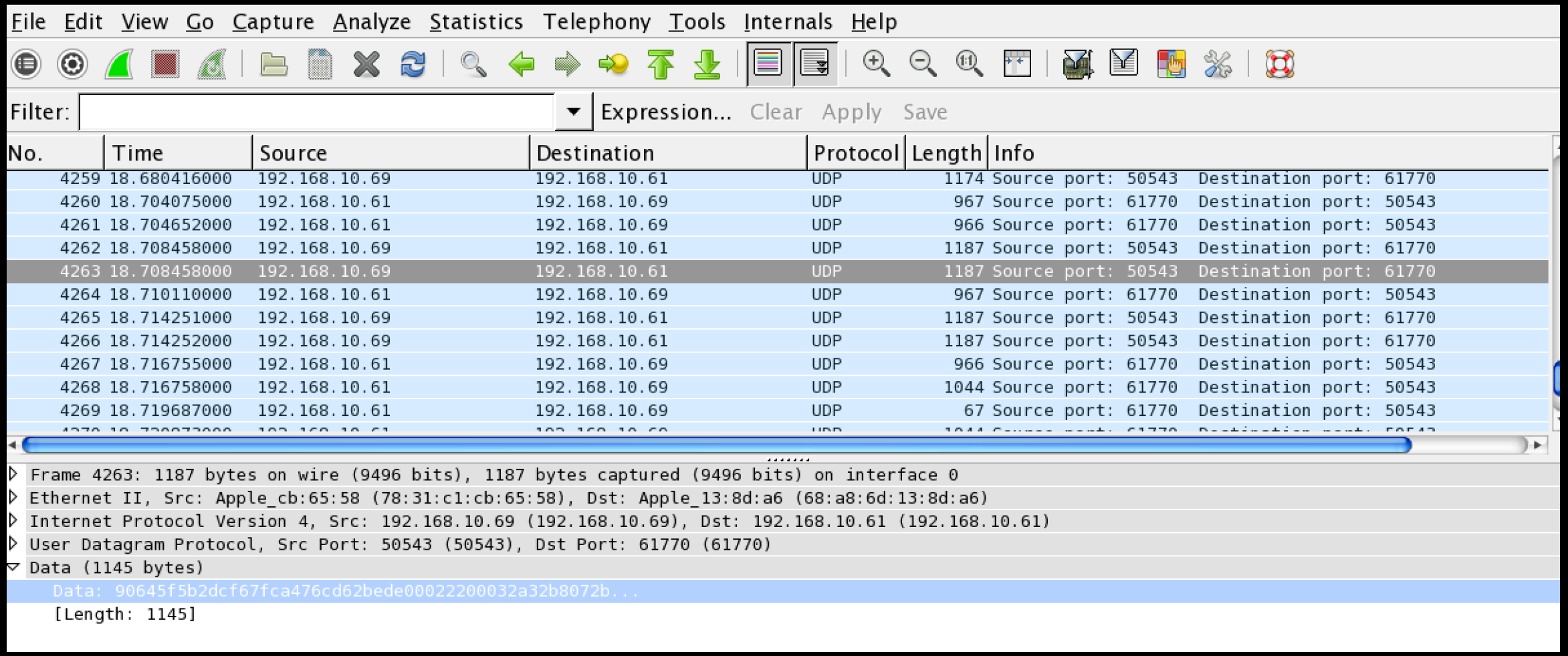
- WebRTC is a free & open project that enables web browsers with Real-Time Communications (RTC) with an easy to use JavaScript API
- Our mission:
  - "To enable rich, high quality, RTC applications to be developed in the browser via simple JavaScript APIs and HTML5."

What is WebRTC?




WebRTC Reference App
WebRTC Reference App
https://apprtc.appspot.com/?r=16400670
Google
Google Maps
YouTube
Wikipedia
News
Popular
Would you like to share your camera and microphone with apprtc.appspot.com?
Camera to share:
FaceTime HD Camera
Microphone to share:
default (Built-in Microphone)
Share Selected Devices

Media Encryption: DTLS-SRTP



Info Tidbits

- Website: www.webrtc.org
- WebRTC project heavily backed by Google, Cisco, Ericsson
- WebRTC provides royalty free codecs
  - Audio codecs: iSAC, iLBC, OPUS
  - Video codec: VP8

Short History

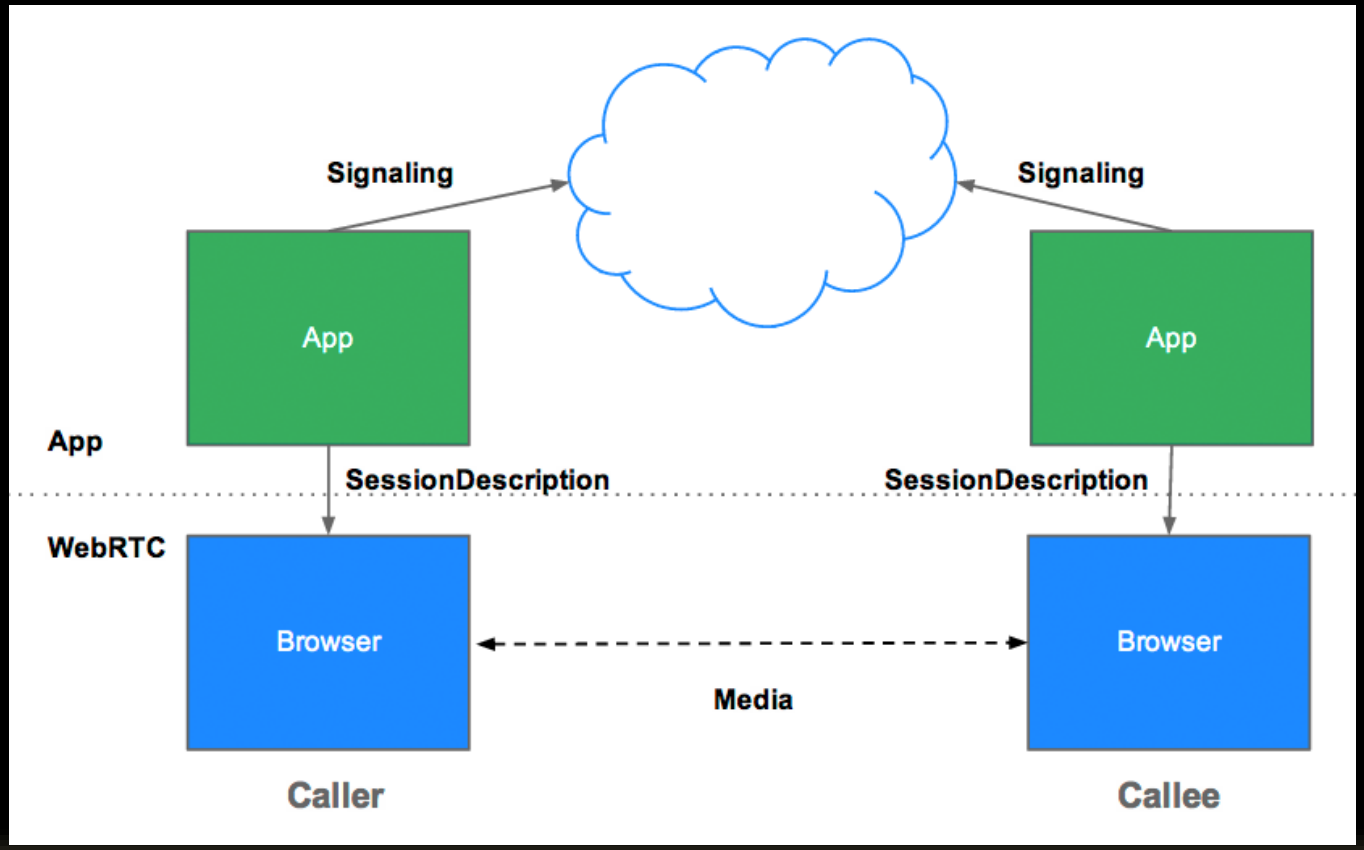
- Google acquired companies in 2010
  - On2
    - Video codec company (VP8)
  - GIPS
    - Audio codecs (iSAC, iLBC)
- Standards push in IETF and W3C
- Microsoft draft specification proposal
- Cisco open sources H.264 in 2013
- Apple Safari?

The Wild West of WebRTC: Signaling

- WebRTC Signaling = Undefined
  - No defined protocol for signaling; just exchange session descriptions objects; then media flows peer-to-peer between web browsers
- Will make things very interesting for how WebRTC applications are deployed
  - Session objects are exchanged, but any method of signaling and transport can be used
  - Some examples of Transports and Messaging Data
    - WebSocket Transport
      - SIP over WebSockets (Messaging)
      - Node.js, socket.io (Two libraries to implement WebSockets)
    - XMLHttpRequest, JSON
    - XMPP

JSEP: JavaScript Session Establishment Protocol

JSEP Architecture



JSEP: JavaScript Session Establishment Protocol

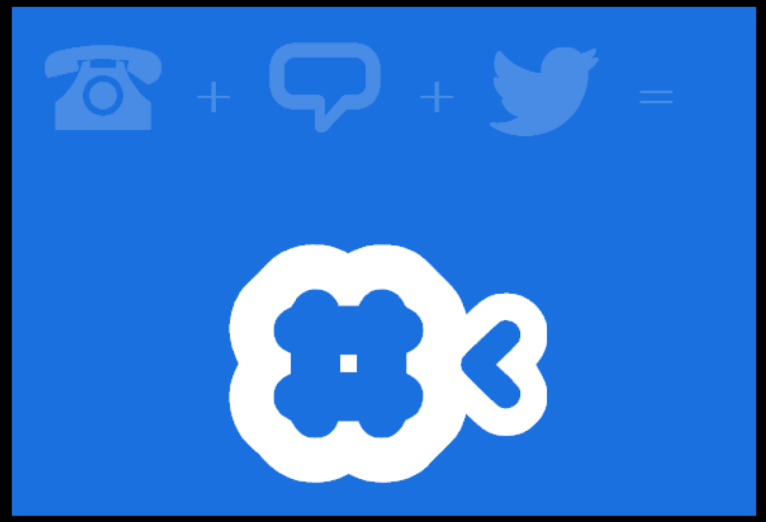
- JSEP requires the exchange between peers of *offer* and *answer*: the media metadata mentioned above. Offers and answers are communicated in Session Description Protocol format (SDP), which look like this:

```
v=0
o=- 7614219274584779017 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video
a=msid-semantic: WMS
m=audio 1 RTP/SAVPF 111 103 104 0 8 107 106 105 13 126
c=IN IP4 0.0.0.0
a=rtcp:1 IN IP4 0.0.0.0
a=ice-ufrag:W2TGCZw2NZHuwlnf
a=ice-pwd:xdQEccP40E+P0L5qTyzDgfmW
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=mid:audio
a=rtcp-mux
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:9c1AHz27dZ9xPI91YNfSlI67/EMkjHHIHORiClQe
a=rtpmap:111 opus/48000/2
…
```

Source:
www.html5rocks.com

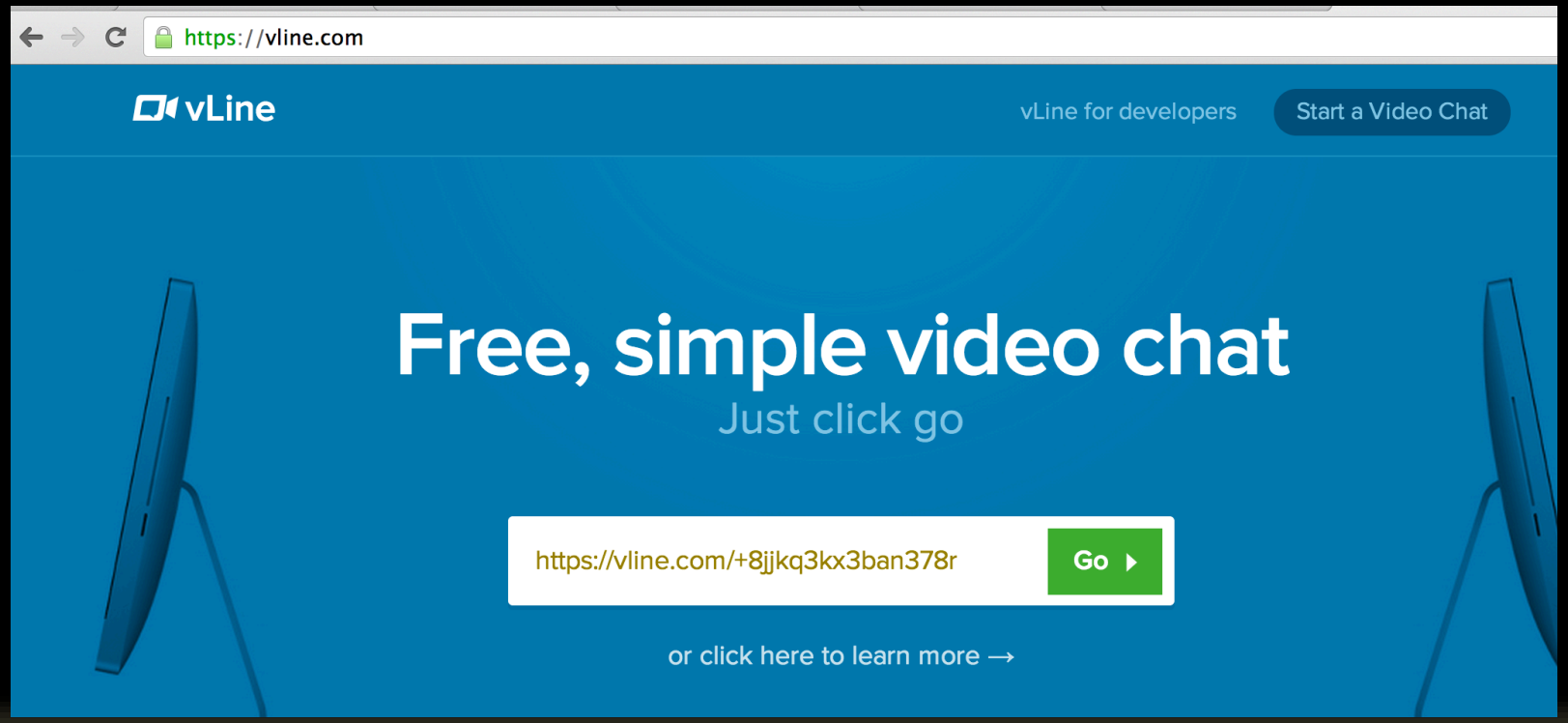
Sample Apps (1)

- Twelephone (Twitter + Telephone)
  - www.telephone.com
  - Example: customer with a problem is reached through twitter; clicks on a link in a tweet and gets voice or video call
  - Uses Node.js for signaling



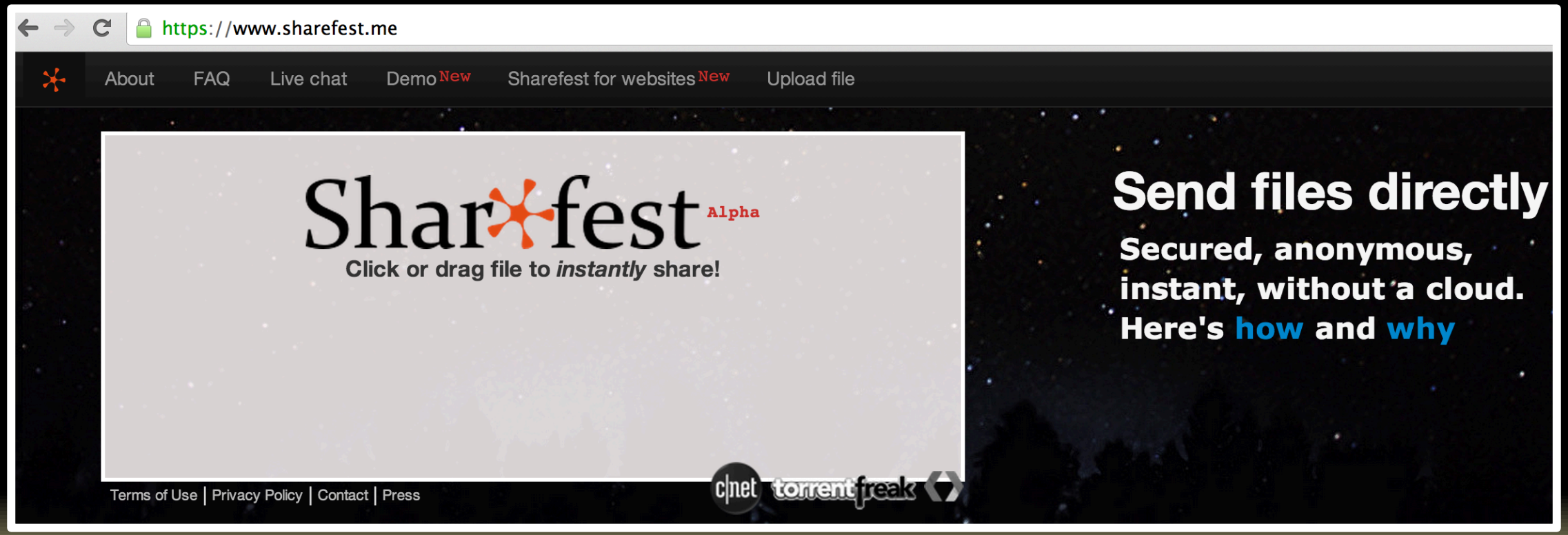
Sample Apps (2)

- VLine
  - www.vline.com



Sample Apps (3)

- Sharefest
  - www.sharefest.me



Security Framework

- Security built in from beginning of WebRTC
  - WebRTC Security Architecture
    - http://tools.ietf.org/html/draft-ietf-rtcweb-security-arch-07
  - Main components of WebRTC Security architecture
    - Mandatory encryption for audio/video media and data channels
    - Secure UI: A user must opt in for camera and microphone control
    - Browser Sandbox

WebRTC Mandatory Encryption

- Mandatory encryption for media
  - DTLS Handshake, following by SRTP encryption
  - Referred to as "DTLS-SRTP", and happens automatically with RTCPeerconnection() offers
  - Provides high degree of assurance for confidentiality, authentication, and prevents replay attacks
- In order to help secure a WebRTC application, it is imperative to use TLS
  - HTTPS instead of HTTP
  - WSS (WebSockets transport with TLS)

WebRTC User Interface

- User Interface security
  - In the Web browser, when the user connects to a WebRTC Application, they must grant explicit permission for the application to use their camera and microphone
  - But there could be issues in the future with persistence

Security Risks and Issues (1)

- Threat: Bugging a user's computer to spy on audio and video, or transfer data
  - A user must give consent to JS application gaining access to their microphone and camera
    - Web browser same origin policy
    - Risk of a user inadvertently clicking on something that gives camera or microphone control
    - Risk of a user being duped into granting access, believing they are granting access to a different website / JS application
  - Malicious JavaScript applications
    - Targeted Spear Phishing via emails containing obfuscated JS
    - Social Engineering
    - Waterhole attacks
    - Any website could have malicious JS code embedded

Security Risks and Issues (2)

- WebSockets
  - WebSockets are an upgraded HTTP connection, that uses a bidirectional TCP socket for web communication
  - Websockets are new, evolving
  - Security issues and concerns for using WS for WebRTC signaling
  - Can the attacker elicit the WS listening port as an unauthenticated user?
  - DoS, DDoS
  - WS + WebRTC is parallel to having a publicly accessible listening SIP service

WebSocket Security Quick Look

- Issues observed
  - Lack of authentication on http before allowing access to WS
  - WebSocket protocol does not specify any authentication or authorization; must be properly implemented into the application
- Simple demonstration of a potential issue in 5 minutes or less

WebSocket Security Tool

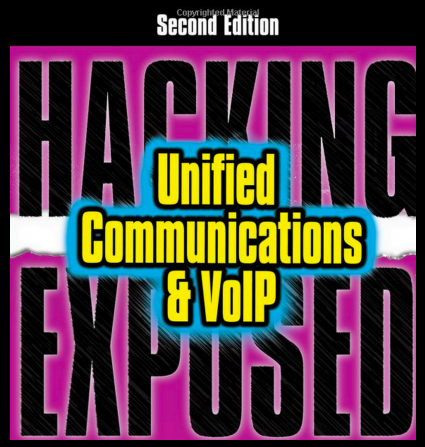
- Security tools already prevalent for testing security of WS
  - https://github.com/koto/socket_io_client

README

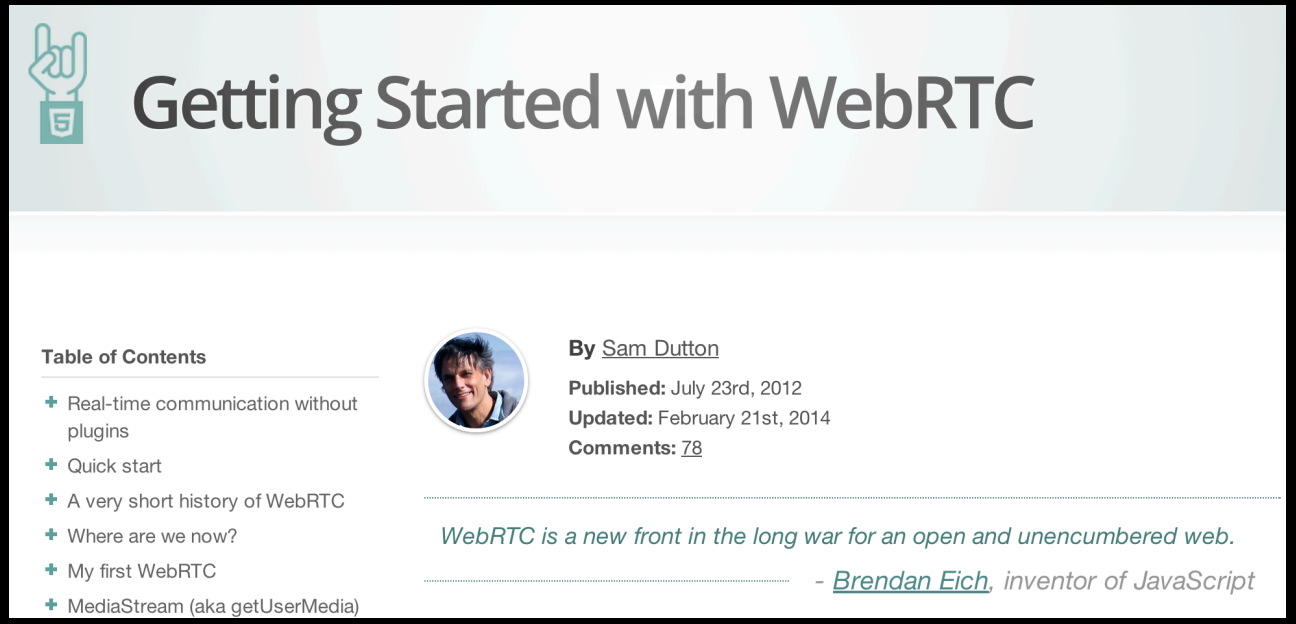
```
A simple malicious Socket.IO client as a Python script.

http://blog.kotowicz.net/2011/03/html5-websockets-security-new-tool-for.html

It can:
 - Handshake with a Socket.io server
 - Ignore all Origin restrictions
 - Transparently handle all socket.io heartbeats
 - Send arbitrary messages - from a prompt or an input file. Messages could be
raw or
   properly formatted according to socket.io protocol
 - Receive/log all server messages
```

More Information

- VoIP Security
  - UC Hacking exposed, 2nd Edition



- WebRTC
  - http://www.html5rocks.com



Contact

- Thank you!
- Jason Ostrom, CCIE #15239
  - Security Consultant, Stora
  - jostrom@storasec.net