Finding VoIP vulnerabilities while you sleep

VoIPER

Format of the talk

- Background info on VoIP and previous research
- Introduction to VoIPER
- Description of some of its features
- Some demos and usage examples
- The results of my testing
- Q&A

About me

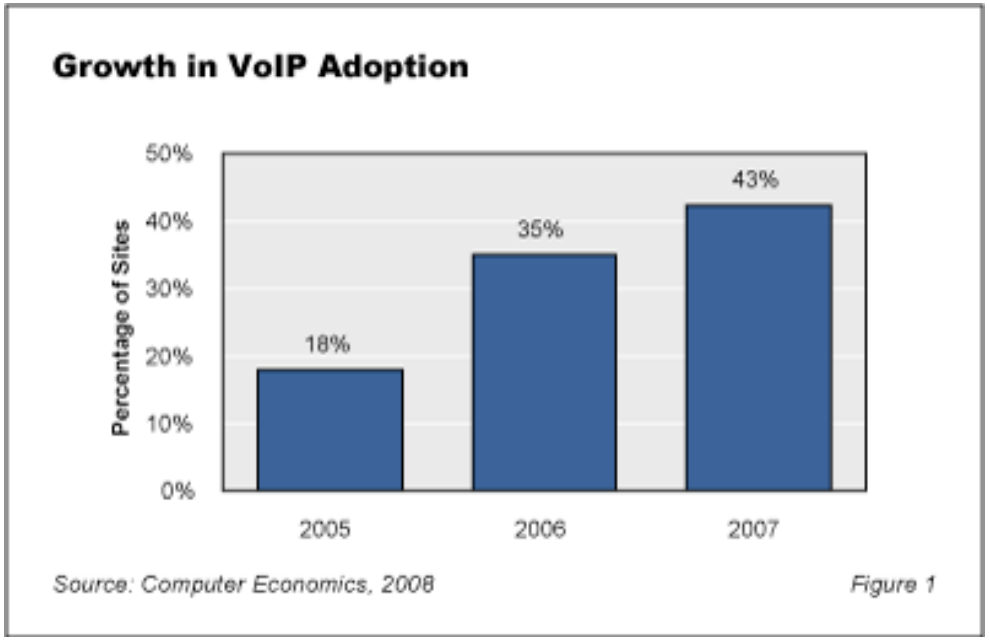
- From Ireland
- Just finished a Bsc in Computer Systems
- About to start a Msc in Computer Science
- Interested in pretty much anything to do with hacking, security and computers in general
- http://www.unprotectedhex.com

What is VoIP

- Using the IP network to route voice data
- Can be used exclusively of the traditional phone network or in tandem
- Variety of devices typically involved
- Many familiar companies from networking and telecoms
  - Cisco
  - Nortel
  - Avaya

Popularity of VoIP

- Seeing steady adoption across the board
- ~50% of large businesses are using it in some form in 2008



Popularity of VoIP

- Why so popular?
  - Reduced costs – Average of 20% reduction
  - Location independence
  - Independence from telcos

What protocols are involved

- SIP
- H.323
- H.225
- H.239
- H.245
- RTCP
- SDP
- MGCP

What protocols are involved

- IAX2
- Skype
- H.460
- H.450
- RTP
- STUN
- RSVP
- SS7
- ….and so on

SIP

- Sponsored by the IETF
- Open standard – RFC 3261
- Similar in format to HTTP
- One of the most popular protocols for consumer devices
- Used for command and control
  - e.g. session initiation and teardown
- Other protocols handle data transfer
  - e.g. RTP

SDP

- Carried as the content in certain SIP requests
- Negotiates the codecs to use for the session
  - Audio
  - Video
  - Extended to cover ‘Fax over IP’ (T.38)
- Human readable
- Combined with SIP, forms an incredibly flexible protocol set

A typical SIP/SDP request

```
INVITE sip:201@192.168.3.102 SIP/2.0
CSeq: 536870905 INVITE
Via: SIP/2.0/UDP 192.168.3.104:6060;branch=z9hG4bKmj1079uq
From: "VoIPER" <sip:201@192.168.3.104:>;tag=hkuybniovshg
Call-ID: jqzedy9kvtrmaw1@TheKlatchianHead
To: "201" <sip:201@192.168.3.102>
Content-Type: application/sdp
Content-Length: 378

v=0
o=- 1190505265 1190505265 IN IP4 192.168.3.104
s=Pwning your SIP
c=IN IP4 192.168.3.104
m=audio 5028 RTP/AVP 101
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

H.323

- ITU sponsored
- Dominant in the backbone of voice networks and large enterprise deployments
- A system specification describing the use of various other protocols including:
  - Registration, Admission and Status (H.225)
  - Call signalling (H.225)
  - Multimedia control and capability determination (H.245)

VoIP: A hackers dream

- Integrates the voice communications of an organisation into an environment the attacker is familiar with
- Same protocols, tools and environments
- Open standards and accessible devices
- Scary as hell when you think about it – you just moved your entire comms infrastructure to our playground
- Cheers!

But it's just a regular phone…

- Not quite …
- Complex operating systems
- Functional TCP/IP stack
- Run a variety of services
- Introduces a whole new attack vector into your network

Possible attack scenarios

- Targetted ‘C’ level attacks
- Botnets
- Eavesdropping
- Worms
- DoS attacks on communications infrastructure
- Every other attack you get on a TCP/IP network

Previous VoIP security research

- Two possible viewpoints
  - Attacking the protocol design
    - Authentication
    - Authorization
    - Encryption
  - Attacking the protocol implementation
    - Aiming to find vulnerabilities leading to DoS or remote code execution

Attacking the design

- Performed in the same fashion as an attack on any network service
- Enumeration, scanning, cracking accounts, MITM attacks, flooding etc.
- Using the last X years experience of attacking TCP/IP based protocols
- Threats can be managed using the same methodologies as any network service

Attacking the design

- Plenty of tools available
- ‘Traditional’ ones such as nmap and co.
- More specialised ones such as SIPVicious and VoIPHopper
- SIPVicious
  - Incorporates tools for mapping a network, finding user accounts and cracking their passwords
- VoIPHopper
  - Jumping between VLANs

Attacking the implementation

- What VoIPER is all about
- A number of other tools available
  - PROTOS
  - KiF
  - INTERSTATE
  - Codenomicon, Mu Dynamics etc
- Generally successful at finding bugs
- A few drawbacks to each

Attacking the implementation

- Some limitations of current tools
  - Closed source or difficult to acquire
  - Difficult to extend or modify
  - Limited test sets
  - Primitive/No support for crash detection and other features required for full automation

VoIPER – Introduction

- Cross platform, open source VoIP fuzzing toolkit
- Currently aimed at the SIP and SDP protocols
- Protocol aware backend that can manage SIP sessions and manipulate the device under test into different states
- Extensive logging, target management and crash recreation tools
- Automation ftw!

VoIPER – Fuzzers

- ~10 ready-to run SIP and SDP fuzzers
- Cover the vast majority of their RFCs and generate well over 200,000 tests
- Fire-and-forget - No protocol knowledge required by the user
- Mapped out using the Sulley Fuzzing Framework
- Delivered using the VoIPER SIP core

VoIPER – Fuzzers

- Mapping:

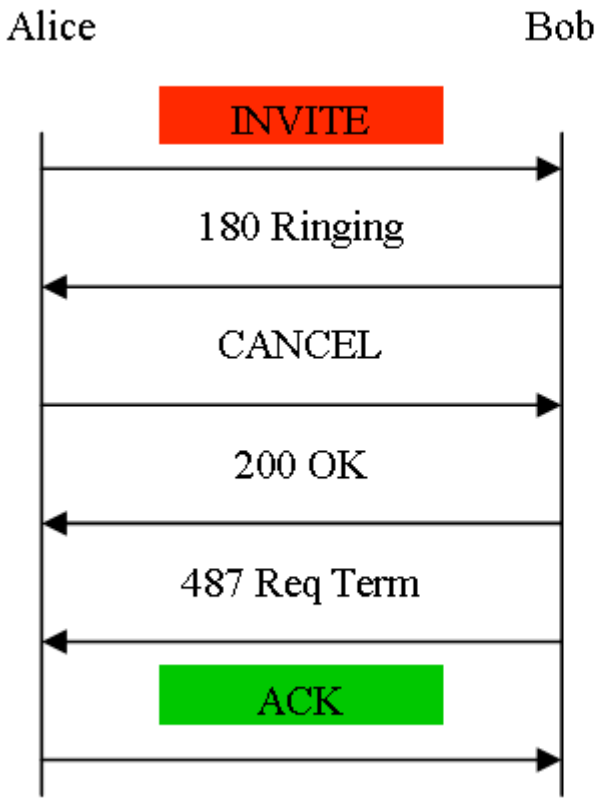
```
s_static("Content-Length: ")
s_dword(512, fuzzable=True,
  format="ascii")
s_static("\r\n")
```

- Example header: Content-Length: -1
- Hilariously enough caused a certain VoIP client to crash

VoIPER – SIP core

- Basic SIP library
  - SIP user agent
  - Transaction management system
  - Collection of pre-made SIP transaction descriptions
- Manipulates the target device into the required state for the test
- Allows the fuzzer to inject fuzz tests into any part of the protocol state

VoIPER – SIP core



VoIPER – SIP core

```
invite_with_cancel_dict =
        {sip_parser.r_SEND :
                (invite.valid,
                 {
                    sip_parser.r_1XX :
                    (cancel.valid,
                     {
                  sip_parser.r_2XX : (None, None),
                  sip_parser.r_4XX : (ack.fuzz, None)
                     }
                    )
                 }
                )
               }
```

VoIPER – Crash detection

- Two types provided
  - Protocol based
  - Process based
- Essential for full automation of the fuzzing process
- Allows for detailed reporting

VoIPER – Target management

- Starting/Restarting the target device
- Minimises the amount of monitoring and interaction required
- Useful against some devices that suffer from chronic DoS syndrome
- Requires a script running on the target device
- Built on components from Sulley

VoIPER – Crash re-creation

- A crash is useless if we can't recreate it on demand!
- Post crash logging should allow automatic recreation of a particular issue
- Process based crash detection provides extra crash info that can help

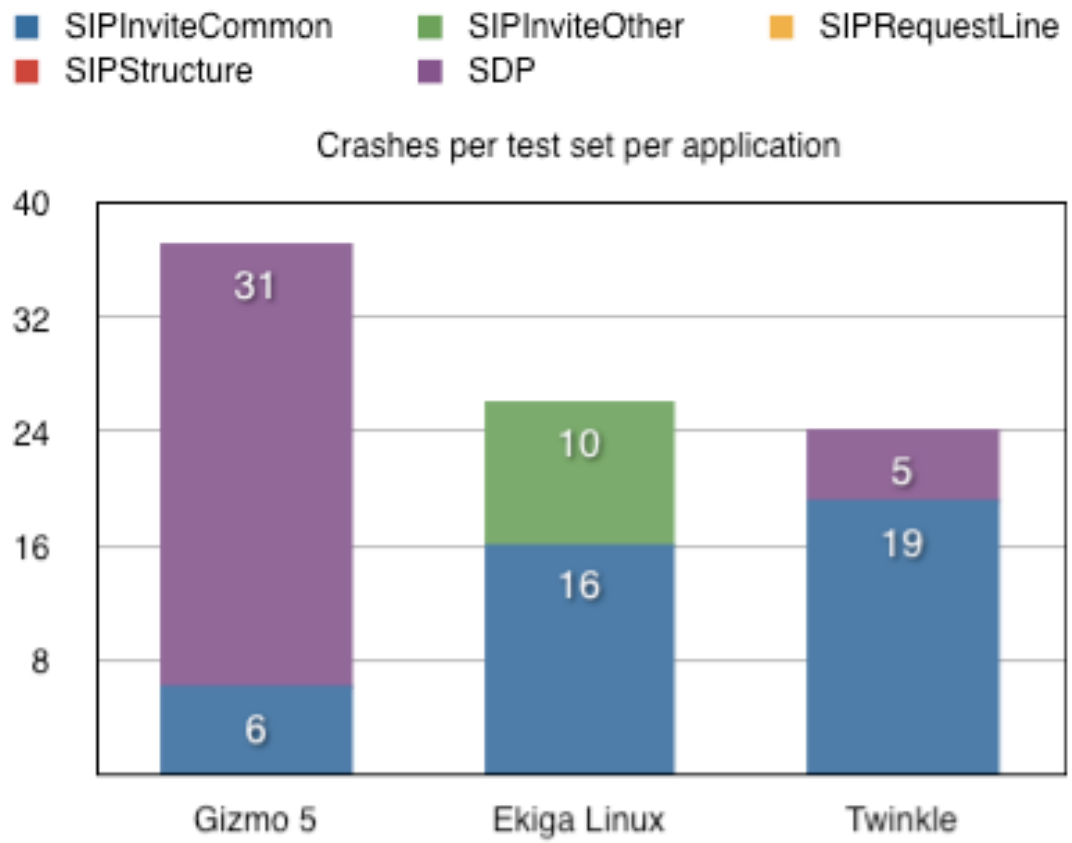
Demo

I’m in ur network breakin ur stuff

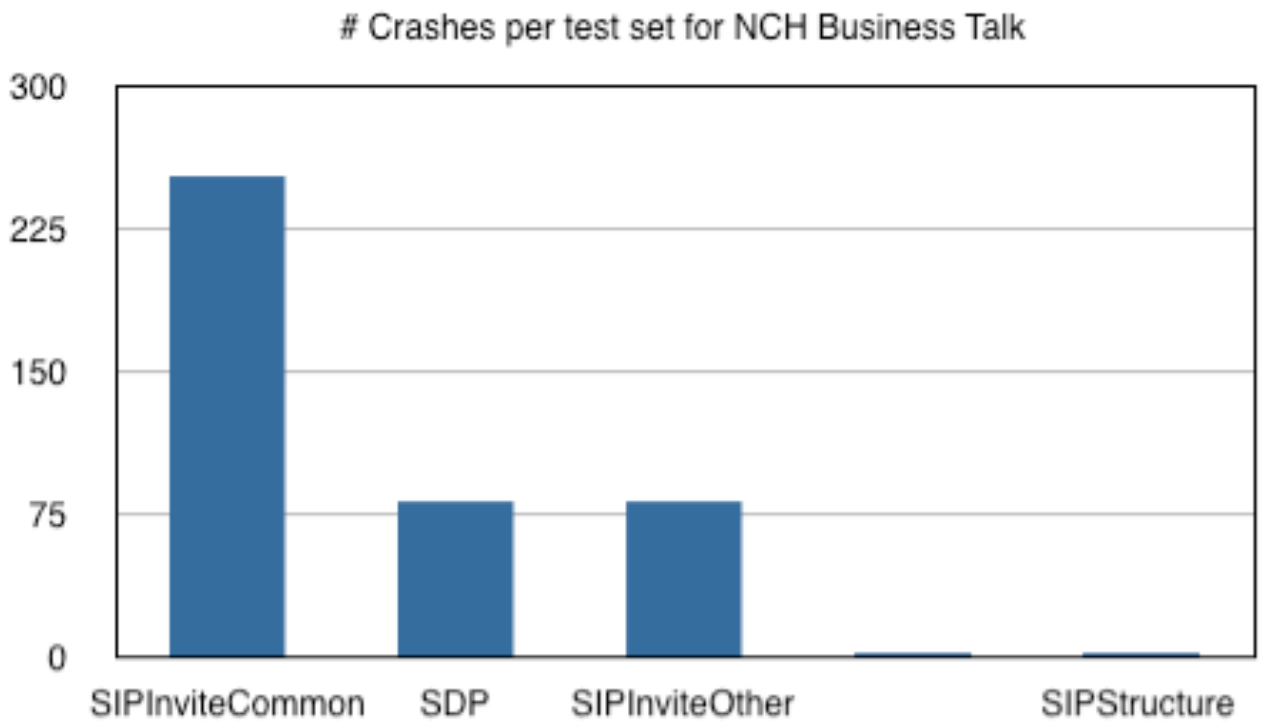
VoIPER - Testing

- So…..did it work?
- Initial testing focused on 4 VoIP clients
  - Ekiga
  - Gizmo5
  - Twinkle
  - NCH Business Talk
- Testing SDP and SIP INVITE processing

Test results



1999 called – It wants its developers back



Testing conclusions

- Plenty of encouragement to test the rest of the SIP protocol and other devices
- Single-packet-o-death crash in all devices
- Types of bugs
  - Null pointer dereferences
  - Memory corruption
- Any product that has the letter ‘NCH’ in its title should be avoided like the plague

VoIPER conclusions – The bad

- Writing a generational fuzzer takes a LOT of time
  - A questionable investment?
  - How many of the bugs found could have been found by a mutational fuzzer, and how quickly?
- Attempting full coverage of a RFC is both unnecessary and ~~batshit~~ moderately insane
  - Many obscure features not implemented in devices tested
  - Most crashes occurred when fuzzing common fields

VoIPER conclusions – The Good

- Fuzzer + crash detection + target management == bug hunting while you sleep
- Possible to start testing essentially any SIP device in a matter of seconds
- Usable by anyone with half a brain
- Easily extendable if it doesn’t do exactly what you want

Thanks

- Terron Williams
- Everyone else who helped out in the beta testing
- Everyone on STS/OTW that helped out

The end

- http://www.unprotectedhex.com
- http://www.unprotectedhex.com/voiper-wiki/
- http://voiper.sourceforge.net

Questions?