Voice over IP Vulnerability Assessment

Ph.D Thesis Defense

Humberto J. Abdelnur

Henri Poincaré University - Nancy 1

Nancy-Université
Université Henri Poincaré

INRIA - Nancy Grand Est

INRIA _ MADYNES

March 30, 2009

Outline

Introduction

1. Background
2. Motivation
3. Challenges

Contributions

4. Fingerprinting
5. Fuzzing
6. Authentication Analysis

Conclusion

7. Summary of Results
8. Future Work

Introduction

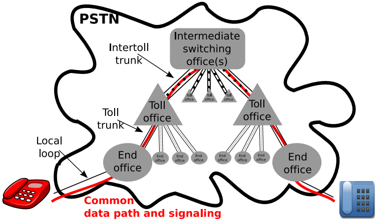
Voice over IP Telephony (VoIP)

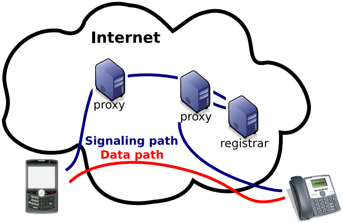
PSTN

- Intelligence concentrated in the network
- Circuit-switched network
- Evolution depends on the core hardware

VoIP

- Intelligence distributed over the equipments
- Packet-switched network
- Evolution depends on software upgrades





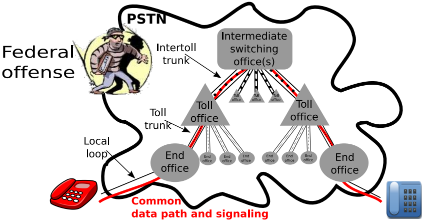
Voice over IP Telephony (VoIP)

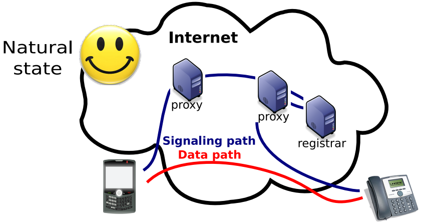
PSTN

- Intelligence concentrated in the network
- Circuit-switched network
- Evolution depends on the core hardware
- Data flows over a closed network

VoIP

- Intelligence distributed over the equipments
- Packet-switched network
- Evolution depends on software upgrades
- Data flows over an open public network





VoIPSA[1] VoIP Threat Taxonomy

- Eavesdropping, interception and modification

  (e.g. rerouting, alteration, hijacking)
- Denial of Service

  (e.g. flooding, network services DoS/DDoS, malformed protocol messages, fake teardown of session)
- Service Abuse

  (e.g bill bypassing, hijacking)
- Social threats

  (e.g. misrepresentation of entities,unwanted contacts, . . .)
- Physical access

  (e.g. social engineering attacks)
- Interruption of services

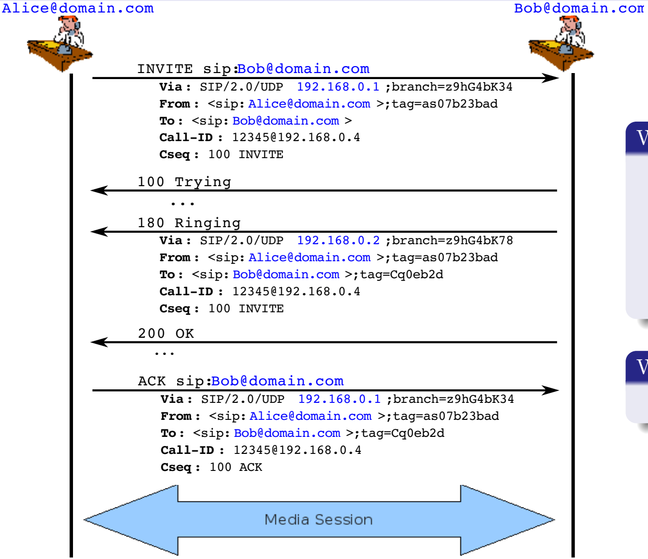
  (e.g. loss of power, resource exhaustion, latency)

[1] VoIP Security Alliance. `http://voipsa.org`

Rethinking the Threats

- No need to use huge resources to perform an attack
- Traffic sniffing is not always required
  - e.g. remote-eavesdropping can be setup by tricky-signalling only
- Operational toll-fraud on VoIP networks is easy to perform
- Standard protocols (here SIP) have weaknesses
- VoIP can serve as a new attack vector

SIP Context



```
INVITE sip:Bob@domain.com
    Via : SIP/2.0/UDP  192.168.0.1 ;branch=z9hG4bK34
    From : <sip: Alice@domain.com >;tag=as07b23bad
    To : <sip: Bob@domain.com >
    Call-ID : 12345@192.168.0.4
    Cseq : 100 INVITE

100 Trying
    ...
180 Ringing
    Via : SIP/2.0/UDP  192.168.0.2 ;branch=z9hG4bK78
    From : <sip: Alice@domain.com >;tag=as07b23bad
    To : <sip: Bob@domain.com >;tag=Cq0eb2d
    Call-ID : 12345@192.168.0.4
    Cseq : 100 INVITE

200 OK
  ...

ACK sip:Bob@domain.com
    Via : SIP/2.0/UDP  192.168.0.1 ;branch=z9hG4bK34
    From : <sip: Alice@domain.com >;tag=as07b23bad
    To : <sip: Bob@domain.com >;tag=Cq0eb2d
    Call-ID : 12345@192.168.0.4
    Cseq : 100 ACK
```

What is SIP?

- A signalling protocol
- Request/Response structure (HTTP-like)
- Stateful
- Text-based protocol

What SIP is not?

- Media transport

Research Challenges

Network Fingerprinting

- Identifies who is the source entity of specific messages
- Assessment, discovery of deployed equipments
- Need of an automation bootstrapping phase
    - SIP is a complex protocol widely implemented

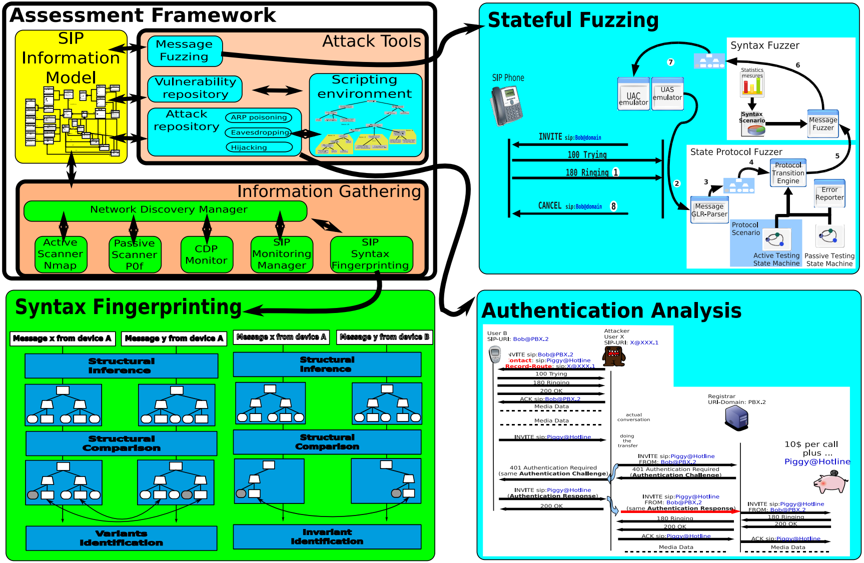
Fuzzing

- Software testing technique to discovery vulnerabilities
- Need to go deeper in the testing
    - SIP is a stateful protocol
- Need for blackbox testing approaches
    - Most implementations are embedded devices

SIP Authentication

- Analysis of the authentication mechanism of SIP

Contributions



Contribution 1:

Fingerprinting

Network Fingerprinting

Objective

- Identify specific devices running a common protocol
- Determine the implementation/vendor from the traffic

Applicability

- Network topology discovery, inventory
- Detect attacks/worms/SPIT systems and stealth intruders

Difficulties

- Banners (if any) can't be trusted
- Signatures can be expressively hidden
- Identify only the significant features
- Complex protocols need to be dealt with

Current Approaches

Active Fingerprinting

- Request/Response queries to observe behavior
- Normal/abnormal messages sent
- Network flow invasive

Passive Fingerprinting

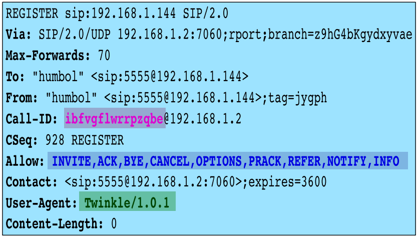
- Monitors and classifies traffic
- Observes syntax, state machine, timing
- "What you see is what you get"
- No overhead traffic generation
- Suitable for "on the fly" fingerprinting

Syntax Signatures

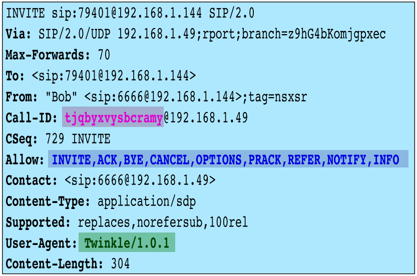
Problem Statement

- Behavior is not fully/specifically documented in RFCs
- Implementations don't fully comply to the specifications

SIP Equipment A



```
REGISTER sip:192.168.1.144 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.2:7060;rport;branch=z9hG4bKgydxyvae
Max-Forwards: 70
To: "humbol" <sip:5555@192.168.1.144>
From: "humbol" <sip:5555@192.168.1.144>;tag=jygph
Call-ID: ibfvgflwrrpzqbe@192.168.1.2
CSeq: 928 REGISTER
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,PRACK,REFER,NOTIFY,INFO
Contact: <sip:5555@192.168.1.2:7060>;expires=3600
User-Agent: Twinkle/1.0.1
Content-Length: 0
```



```
INVITE sip:79401@192.168.1.144 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.49;rport;branch=z9hG4bKomjgpxec
Max-Forwards: 70
To: <sip:79401@192.168.1.144>
From: "Bob" <sip:6666@192.168.1.144>;tag=nsxsr
Call-ID: tjqbyxvysbcramy@192.168.1.49
CSeq: 729 INVITE
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,PRACK,REFER,NOTIFY,INFO
Contact: <sip:6666@192.168.1.49>
Content-Type: application/sdp
Supported: replaces,norefersub,100rel
User-Agent: Twinkle/1.0.1
Content-Length: 304
```

Header Order **Call-ID Length** **Allow Order** **User-Agent Banner**

Syntax Signatures

Problem Statement

- Behavior is not fully/specifically documented in RFCs
- Implementations don't fully comply to the specifications

SIP Equipment B

```
REGISTER sip:192.168.1.144 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.20:5060;branch=z9hG4bK4205b326
From: <sip:7940@192.168.1.144>;tag=000b46d9cb-1a84cfd8
To: <sip:7940@192.168.1.144>
Call-ID: 000b46d9-cb860003-66d2804f-527006cb@192.168.1.20
Max-Forwards: 70
CSeq: 102 REGISTER
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:7940@192.168.1.20:5060;transport=udp>;
Content-Length: 0
Expires: 3600
```

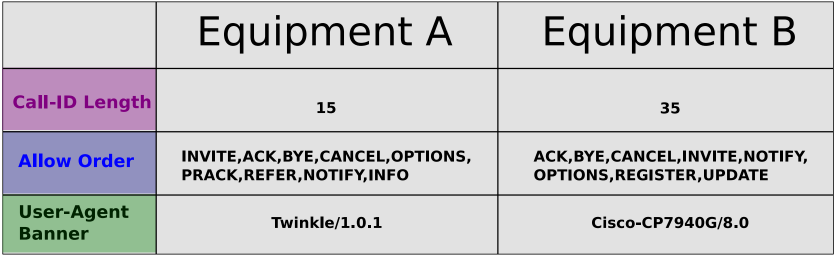
```
INVITE sip:611@192.168.1.144 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.49:5060;branch=z9hG4bK50979e8b
From: "6666" <sip:6666@192.168.1.144>;tag=001ae2bc8b-4f6a3bc6
To: <sip:611@192.168.1.144>
Call-ID: 001ae2bc-8b7c001a-40b4297e-1611ee91@192.168.1.49
Max-Forwards: 70
CSeq: 102 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:6666@192.168.1.49:5060;transport=udp>
Expires: 180
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 276
```

~~Header Order~~ Call-ID Length Allow Order User-Agent Banner

Syntax Signatures

Problem Statement

- Behavior is not fully/specifically documented in RFCs
- Implementations don't fully comply to the specifications



	Equipment A	Equipment B
Call-ID Length	15	35
Allow Order	INVITE,ACK,BYE,CANCEL,OPTIONS, PRACK,REFER,NOTIFY,INFO	ACK,BYE,CANCEL,INVITE,NOTIFY, OPTIONS,REGISTER,UPDATE
User-Agent Banner	Twinkle/1.0.1	Cisco-CP7940G/8.0

Relevant Work

“Incorporating Active Fingerprinting into SPIT Prevention Systems” [7]

- Signatures over the content message
- Manually identified (≈10 devices)
- Not scalable

“Catching the Picospams” [8]

- Automated signature identification
- Source per message identification
- Signatures over natural language written sentences

“Fig: Automatic Fingerprint Generation” [9]

- Automated queries generation
- Queries discrimination based on different observed behavior
- Active approach

“Network Protocol System Fingerprinting” [10]

- State machine induction by traces
- Identify the source by following the transitions

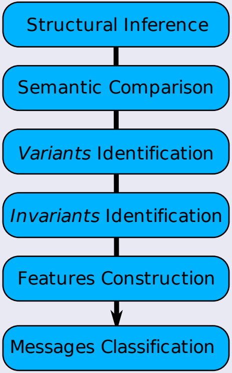
The Big Picture

Challenges[1]

- Be robust to malicious scrubbers
- Identify the source for each message
- Automate signature discovery

[1] Assuming we know the protocol

Operational Framework



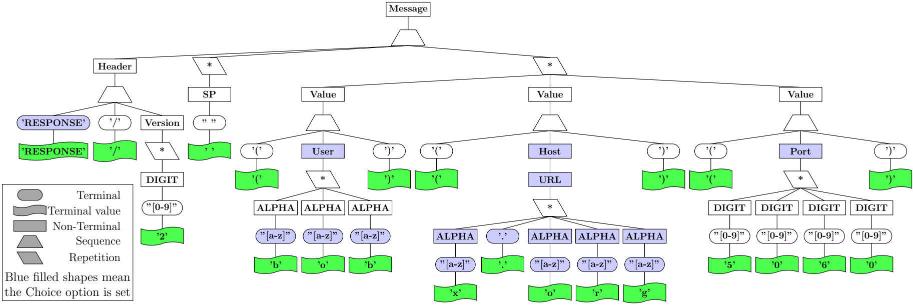
H. Abdelnur, R. State and O. Festor.
"Advanced Network Fingerprinting".
Recent Advances in Intrusion Detection, RAID 2008.

H. Abdelnur, R. State and O. Festor.
"Advanced Structural Fingerprinting in SIP" Live demo.
Principles, Systems and Applications of IP Telecommunications, IPTcomm 2008.

Syntax Inference

- ABNF grammar specification is known
- Messages can be represented by a tree structure
- Structure used rather than just lexicon
- Generic approach, allows the parsing of any rule of any grammar

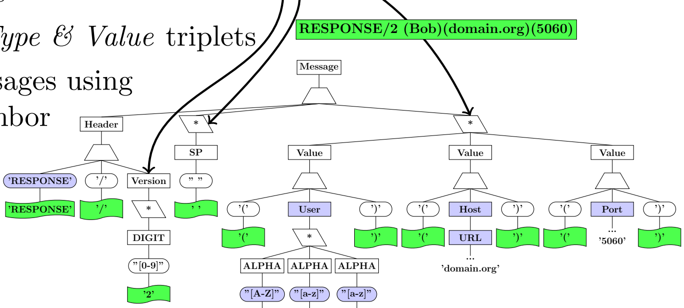
RESPONSE/2 (bob)(x.org)(5060)



Structural Features

- Identify signatures in the paths of the tree
- Identify signatures as:
    - Contents
    - Lengths
    - Orders
    - Functions
- Save *Path, Type & Value* triplets
- Classify messages using Closest Neighbor

Field path	Feature associated	
	Type	Value
Message.2.(2)	Order	User, Host, Port
Message.1.(?)	Length	1
Message.0.Header.2	Content	'2'



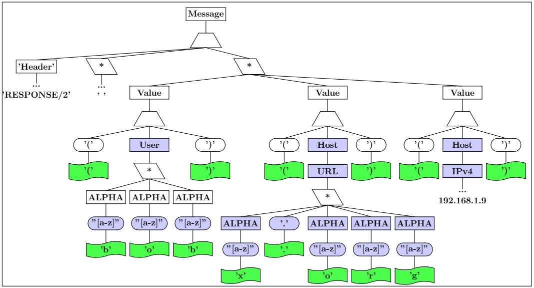
Node Comparison

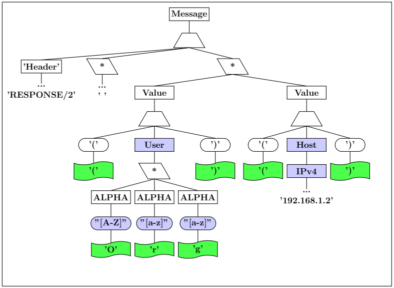
Comparison Matching

- Shared items between nodes
- Tags and ancestors tags must be equal
- Sequences children must be ordered equally
- Repetitions can be unordered

RESPONSE/2 (bob) (x.org) (192.168.1.9)

RESPONSE/2 (Org) (192.168.1.2)

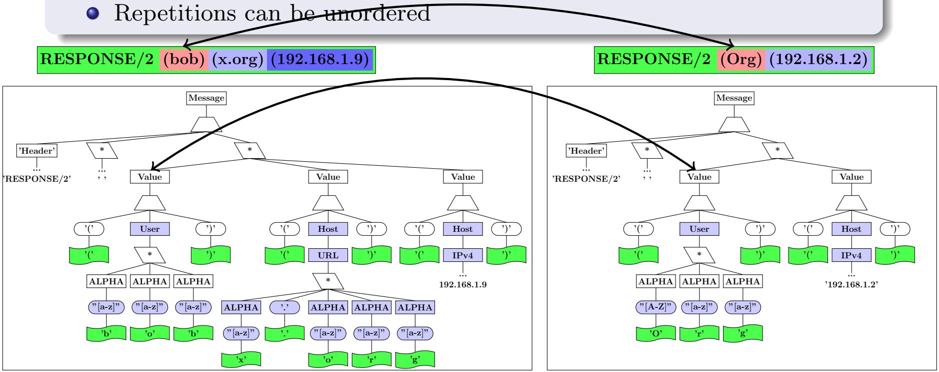




Node Comparison

Comparison Matching

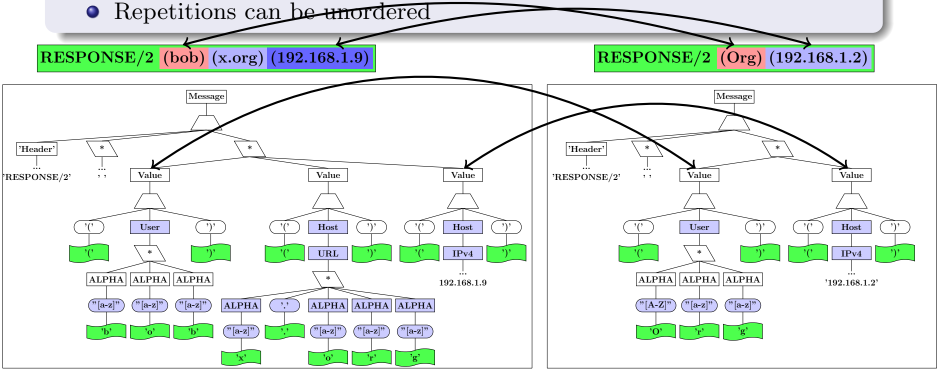
- Shared items between nodes
- Tags and ancestors tags must be equal
- Sequences children must be ordered equally
- Repetitions can be unordered



Node Comparison

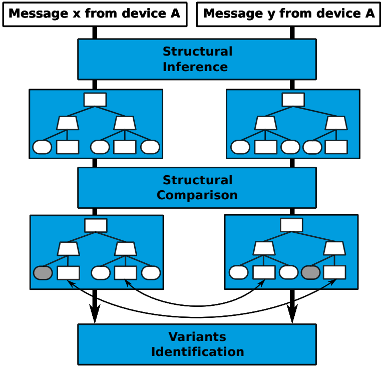
Comparison Matching

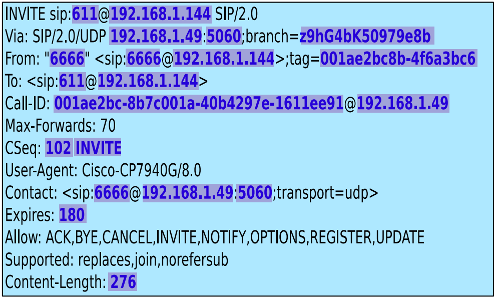
- Shared items between nodes
- Tags and ancestors tags must be equal
- Sequences children must be ordered equally
- Repetitions can be unordered



Phase 1: Variants Identification

- Pairwise comparison of messages from the **same device**
- The differences identify the *Variant fields*
- These fields are of no interest
    - Configuration values
    - Context specific values

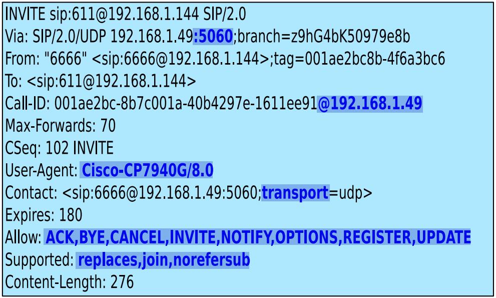




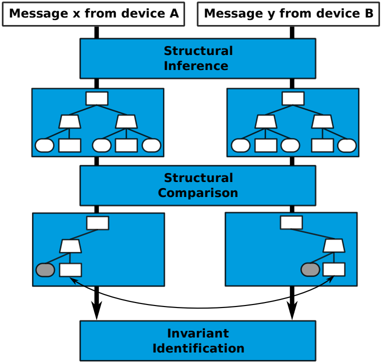
```
INVITE sip:611@192.168.1.144 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.49:5060;branch=z9hG4bK50979e8b
From: "6666" <sip:6666@192.168.1.144>;tag=001ae2bc8b-4f6a3bc6
To: <sip:611@192.168.1.144>
Call-ID: 001ae2bc-8b7c001a-40b4297e-1611ee91@192.168.1.49
Max-Forwards: 70
CSeq: 102 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:6666@192.168.1.49:5060;transport=udp>
Expires: 180
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 276
```

Phase 2: Features Identification

- Pairwise comparison of messages from **different devices**
- Filter differences that are *Invariant fields*
- These fields are the *Signatures*
  - Same values for the same device
  - but different within implementations



```
INVITE sip:611@192.168.1.144 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.49:5060;branch=z9hG4bK50979e8b
From: "6666" <sip:6666@192.168.1.144>;tag=001ae2bc8b-4f6a3bc6
To: <sip:611@192.168.1.144>
Call-ID: 001ae2bc-8b7c001a-40b4297e-1611ee91@192.168.1.49
Max-Forwards: 70
CSeq: 102 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:6666@192.168.1.49:5060;transport=udp>
Expires: 180
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 276
```



Experimental Results

- Fingerprinting framework implemented in Python
- 21981 recollected SIP messages labeled (from 26 different apps/conf)
- 15% of the messages were sufficient to train the system
- 271 features discovered
- Classifications:
  - Used between 10 to 58 features
  - Average classification time 0.06 seconds

Efficiency

150[1] *Xeon-Woodcrest nodes, dual-core 64 bits, 2GB RAM*

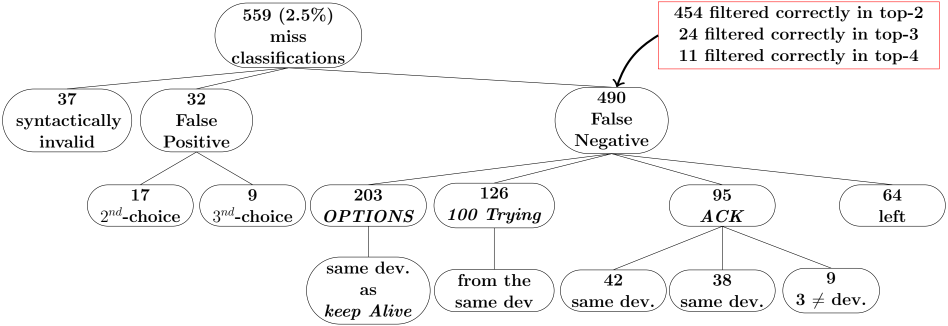
Actions	computed actions	Total time
Phase 1	571.234	1 hour
Phase 2	8.175.419	10 hours

Accuracy

Classification	**True Positive** 21422	**False Positive** 32
	False Negative 490	**True Negative** N.A.
Accuracy 0.998	**Sensitivity** 0.976	**Specificity** 0.999

[1] Experiments were carried out using the Grid'5000 experimental testbed

Error Analysis



454 filtered correctly in top-2
24 filtered correctly in top-3
11 filtered correctly in top-4

559 (2.5%) miss classifications

37 syntactically invalid

32 False Positive

17 2^{nd}-choice

9 3^{nd}-choice

490 False Negative

203 ***OPTIONS***

same dev. as ***keep Alive***

126 ***100 Trying***

from the same dev

95 ***ACK***

42 same dev.

38 same dev.

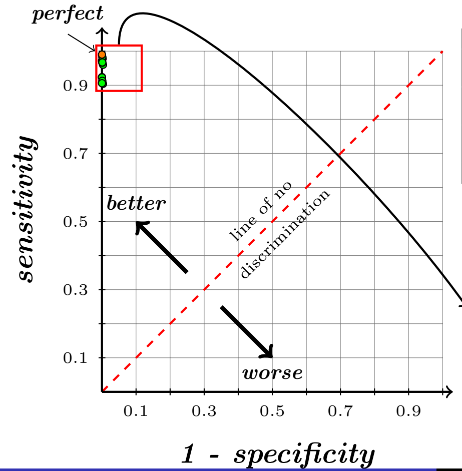
9 3 ≠ dev.

64 left

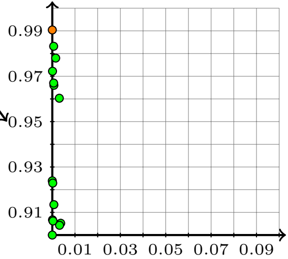
```
OPTIONS sip:192.168.1.4:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.101;rport;branch=z9hG4bKc0a801650000000b4550c64f000000f5000000e
Content-Length: 0
Call-ID: F28A8FE4-1FF9-4937-AF8D-81B29FD607FE@192.168.1.101
CSeq: 20 OPTIONS
From: <sip:0231555777@192.168.1.4>;tag=1286870423922
Max-Forwards: 70
To: <sip:192.168.1.4:5060>
```

Scalability

- 2091 recollected SIP messages (6 different applications)
- Trained several times with only 15% of the traces



Msgs	Feat.	FP	FN	Acc.
15%	125 ∼ 189	3 ∼ 23	35 ∼ 231	0.997 ∼ 0.979
20 ∼ 40%	133 ∼ 194	6 ∼ 24	18 ∼ 44	0.998 ∼ 0.995
50 ∼ 90%	165 ∼ 193	1 ∼ 2	20 ∼ 19	0.998
100%	174	1	20	0.998



Summary

- We built a robust automated signature discovery framework that:
  - Does not rely on lexicon
  - Exploit arborescent structures
  - Is generic
- It was successfully applied to SIP
  - We have a large database of device traces
  - Accuracy of the system is convincing
- Limitations:
  - Syntactically known protocols
  - Clear text flows

Contribution 2:

Fuzzing

INRIA MADYNES

Fuzzing

- Emerged as a branch of Software Testing
- Functional verification is marginal
- Main objective is to find possible potential vulnerabilities
- Important topic for **Development Cycle/Independent Assessment**
- Based on input data validation
  - Random or invalid characters (not too random actually)
  - Malicious data (e.g. string formatters)

Relevant work

Mini-Simulation Toolkit [13]

- Send malformed messages to the target
- Limited data generation

SnooZe[11], Sulley[15]

- Framework for messages generation
- Requires more specification as more precise it gets

GPF[12], Sidewinder[14]

- Evolutionary methods to generate messages
- Hard to estimate what will be the generated output/expected answer

In-Depth Testing of Web Applications[16]

- Replay traces to get deeper in the test
- There is no knowledge between right/wrong transitions (i.e. stateless)

Generally

- Success evaluation depends on crashed or NOT-crashed
- Past events are not considered
- Unable to decide when to stop
  - Time of testing
  - Quantity of tests or some new metrics?

What to Fuzz?

Syntax fuzzing

- Invalid messages may reveal vulnerabilities
- Consider which items of the message should be fuzzed
- Headers or input values may be fuzzed
- Which values may be replaced
- New values may or may not be syntactically correct

Stateful fuzzing

- Unexpected messages may reveal vulnerabilities
- Decide what type of message to send
- Decide when to send the next message

H. Abdelnur, R. State and O. Festor.
"KiF: A stateful SIP fuzzer".
Principles, Systems and Applications of IP Telecommunications, IPTcomm 2007.

H. Abdelnur, R. State and O. Festor.
"SIPping your network".
East coast hacker convention, ShmooCon 2008.

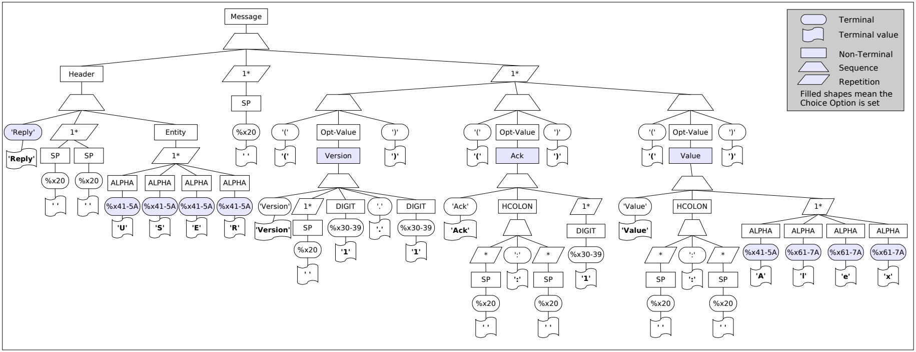
H. Abdelnur, R. State and O. Festor.
"Fuzzing for vulnerabilities in the VoIP space".
European Expert Group for IT-Security, EICAR 2008.

Making Things Easier

- Each protocol has its own grammar specification (e.g ABNF grammars as defined in RFC 2234).
  Why not reuse it?
- Full and precise description of the Protocol Syntax
- Generic approach, allows Parsing & Fuzzing to any Rule of any Grammar

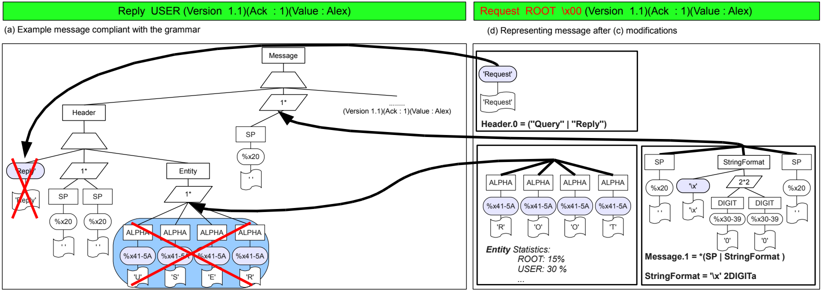
```
Message   =  Header 1*SP 1*( "(" Opt-Value ")" )
Header    =  ("Query" / "Reply") 1*SP Entity
Opt-Value =  (Ack / Value / Version)
Entity    =  1*ALPHA
Ack       =  "Ack" HCOLON 1*DIGIT
Value     =  "Value" HCOLON 1*ALPHA
Version   =  "Version" 1*SP DIGIT ”.” DIGIT
ALPHA     =  %x41-5A / %x61-7A                ; A-Z / a-z
DIGIT     =  %x30-39                          ; 0-9
HCOLON    =  *SP ":" *SP
SP        =  %x20                             ; space
```

Reply USER (Version 1.1)(Ack : 1)(Value : Alex)



Syntax Modifications

- Any grammar rule may be generated (i.e. generation from scratch)
- Statistic measures may influence the reduction (i.e. learning from the past)
- Any existing reduction may be replaced (i.e. mutation or merging)
- New rules can be defined on the fly (i.e. evolving rules)
- Semantic computation may be applied from other nodes (e.g. checksum computations)



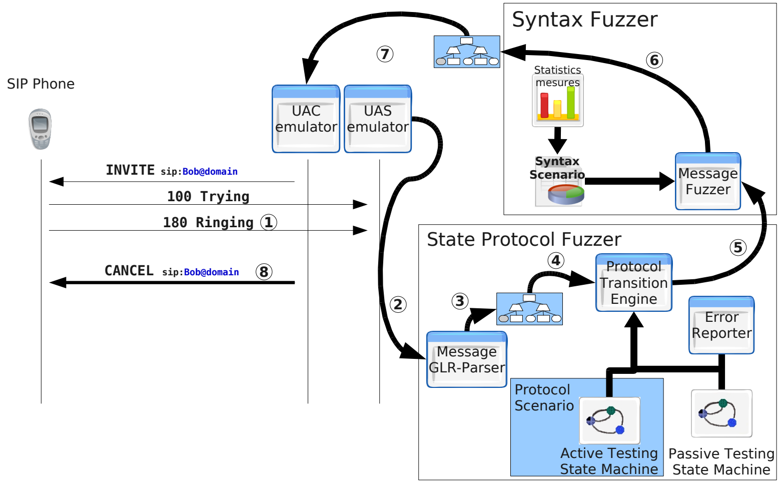
(a) Example message compliant with the grammar

(d) Representing message after (c) modifications

(b) Infered structure from the Message in (a)

(c) Structure modifications to the Message in (b)

KiF Framework Overview

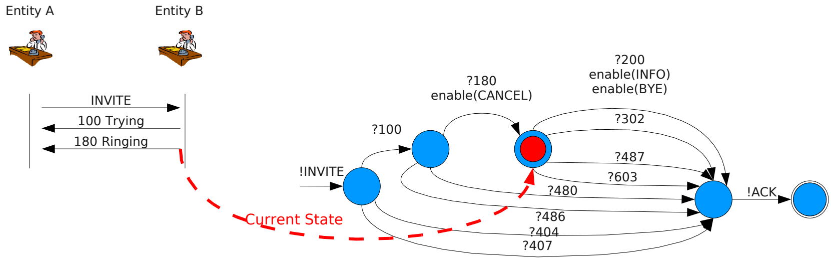


INRIA MADYNES

Behavioral testing

Passive Testing

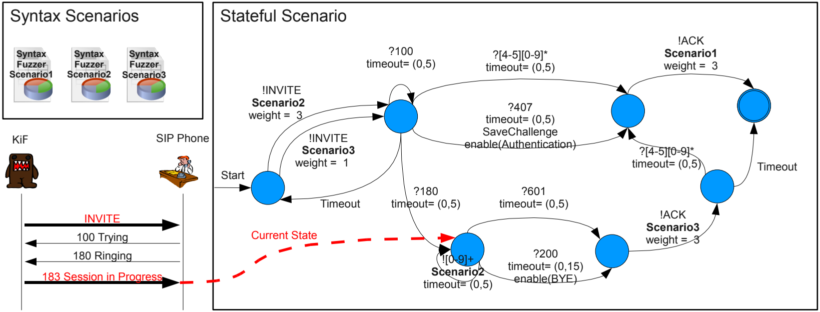
- Collect traces under normal conditions to deduce normal behavior (NP-hard[17])
- Just observes the current traffic
- Infers current state of the unit under test
- Detects abnormal events



Behavioral Testing

Active Testing

- Leads the target into a specific state
- Specify which action must be taken at each step
- Event-Driven Probabilistic Finite Automata based Scenarios



Errors Reporting Conditions

- Syntactically incorrect messages
- No existent passive state machine transitions
- Unexpected message in the current scenario(i.e. when the scenario is trying to avoid the normal protocol flow, e.g. for authenticating)
- Unresponding device

Summary

- Precise and specific fuzzing approach
- Dynamic, results will always be different
- Adaptability & stateful
    - We reach deeper surface of testing
- Tested it in more than 15 leading market devices
    - All implementations present vulnerabilities!

Contribution 3:

Authentication Analysis

SIP Authentication Mechanism

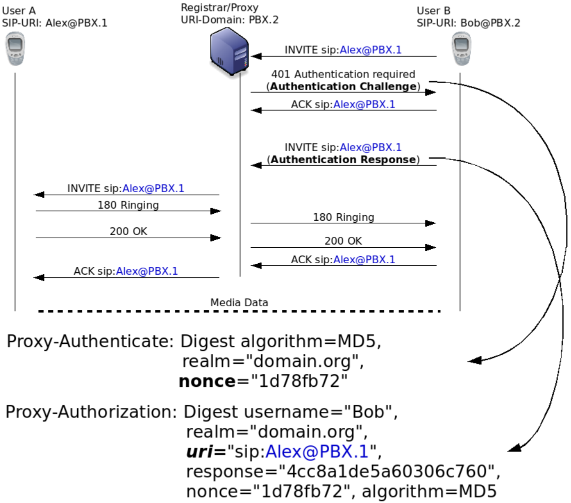
Background

- Authentication based on the design of HTTP
- Shared secret model
- SIP is one of the longest specification made by IETF
- More than 60 extensions exists in the IETF

Objective

- Design fuzzing cases for bypassing authentication
- Analysis of possible failures

SIP Authentication Background[2]



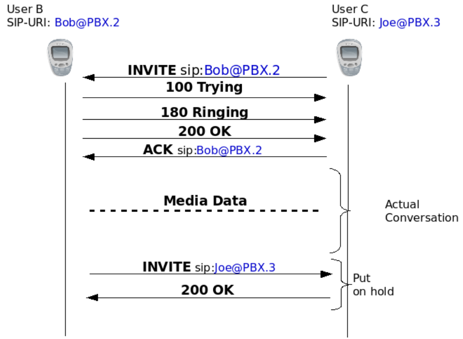
Proxy-Authenticate: Digest algorithm=MD5,
realm="domain.org",
nonce="1d78fb72"

Proxy-Authorization: Digest username="Bob",
realm="domain.org",
uri="sip:Alex@PBX.1",
response="4cc8a1de5a60306c760",
nonce="1d78fb72", algorithm=MD5

[2] RFC-3261, RFC-2617

Re-INVITE feature in SIP

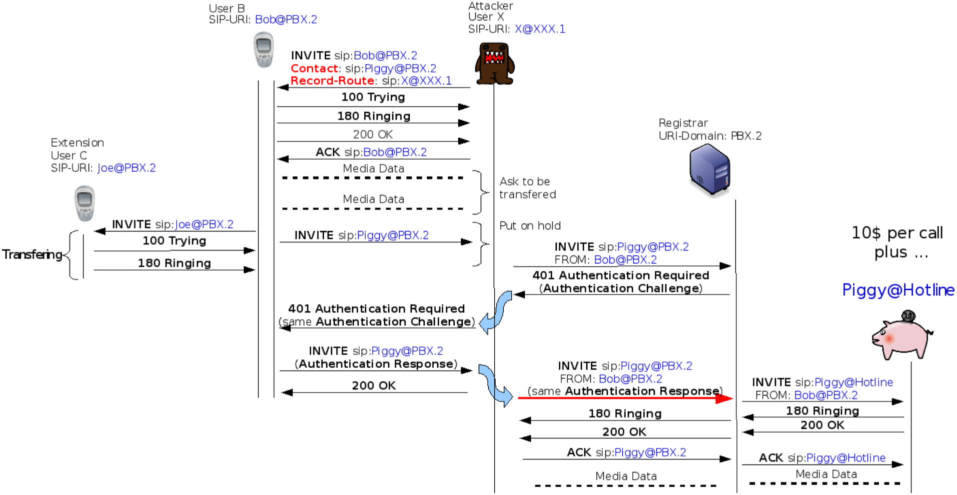
- How re-INVITEs work:



- Can we ask to authenticate re-INVITEs?

Toll-fraud: a SIP Design Flaw

- We may use such authentication at will :)



User B
SIP-URI: Bob@PBX.2

Attacker
User X
SIP-URI: X@XXX.1

INVITE sip:Bob@PBX.2
Contact: sip:Piggy@PBX.2
Record-Route: sip:X@XXX.1

100 Trying

180 Ringing

200 OK

ACK sip:Bob@PBX.2

Media Data

Media Data

Ask to be transfered

Registrar
URI-Domain: PBX.2

Extension
User C
SIP-URI: Joe@PBX.2

Transfering

INVITE sip:Joe@PBX.2

100 Trying

180 Ringing

INVITE sip:Piggy@PBX.2

Put on hold

INVITE sip:Piggy@PBX.2
FROM: Bob@PBX.2

401 Authentication Required
(Authentication Challenge)

401 Authentication Required
(same Authentication Challenge)

INVITE sip:Piggy@PBX.2
(Authentication Response)

200 OK

INVITE sip:Piggy@PBX.2
FROM: Bob@PBX.2
(same Authentication Response)

180 Ringing

200 OK

ACK sip:Piggy@PBX.2

Media Data

10$ per call plus ...

Piggy@Hotline

INVITE sip:Piggy@Hotline
FROM: Bob@PBX.2

180 Ringing

200 OK

ACK sip:Piggy@Hotline

Media Data

Summary

- It sets up a MiM attack without been in the middle
- Flaw based on using the same Method name for different action

Authentication response

```
A1   = username ":" realm ":" passwd
A2   = Method ":" Digest-URI
resp = MD5(MD5(A1) ":" nonce ":" MD5(A2))
```

- Solved by changing the SIP method from "INVITE" to "REINVITE"
- Evaluating the perturbation for SIP

H. Abdelnur, T. Avanesov, M. Rusinowitch and R. State.
"Abusing SIP Authentication".
Information Assurance and Security, 2008, ISIAS '08.

R. State, O. Festor, H. Abdelnur, V. Pascual, J. Kuthan, R. Coeffic, J. Janak, J. Floroiu
"SIP digest authentication relay attack".
The Internet Engineering Task Force, IETF.

Conclusions

Conclusions

Assessment Architecture

- Integrated framerwork for VoIP assessment

Fuzzing

- A complete & specific syntax fuzzer
- Stateful approach design
- Active & Passive testing merged

Network Fingerprinting

- Successful automated syntax signatures discovery
- Generic approach based on the syntax structure

Authentication Analysis

- Mayor vulnerability in the SIP authentication method

Industrial Impact

KiF: a Stateful SIP Fuzzer tool

- 40K lines of code
- Users
  - Georgia Tech, Alcatel-Lucent, Orange Telecom, British Telecom, IPtel, NEC, ...
- Free project

FiF: a Structural Passive Fingerprinting tool

- 15K lines of code
- Patent over the methods

Security Advisories

Common Vulnerabilities and Exposures (CVE)

- Database of publicly known security vulnerabilities and exposures
- Vulnerabilities are reviewed before being added

CVE's list

- Responsible disclosure policies
- More than 15 CVEs disclosed (3 months period)
  - DoS
  - Toll-fraud
  - Remote eavesdropping

Publications

International Conferences

- Recent Advances in Intrusion Detection **(RAID 08)**, Boston, USA. 25% acceptance
- Principles, Systems and Applications of IP Telecommunications **(IPTComm 07)**, New York, USA.
- European Expert Group for IT-Security, **(EICAR 08)**, Laval, France.
- Integrated Management **(IM 07)**, Munich, Germany. 31 % acceptance
- Information Assurance and Security **(IAS 08)**, Naples, Italy.

Security Convention

- **Shmoocon 2008**, annual East coast hacker convention, Washington, USA.

Popular Magazine

- **MISC Magazine** - Edition française: Multi-System & Internet Security Cookbook. Misc #39

Internet Engineering Task Force (IETF) Draft Proposals

- The Common Log File (CLF) format for SIP
- SIP digest authentication relay attack

Future Work

Fingerprinting

- Measure entropy of the fields
- Recognize behavior of protocol stacks
- Learn signatures from unknown protocols

Fuzzing

- Linking testing techniques with fuzzing
- ALCATEL-LUCENT/INRIA joint labs:
  - Extend KiF to be SIP independent
- ANR Project VAMPIRE:
  - Evaluation of optimal fuzzing strategies
  - Virtualisation instrumentation
  - Closed-loop fuzzing

Questions

&

Answers

[1] J. F. Ransome and J. Rittinghouse.
“Voice over Internet Protocol (VoIP) Security”.
Digital Press. Newton, USA.

[2] P. Thermos and A. Takanen.
“Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures”.
Addison-Wesley Professional.

[3] T. Porter and J. Kanclirz and A. Zmolek and A. Rosela and M. Cross and L. Chaffin and B. Baskin and C. Shim.
“Practical VoIP Security”.
Andrew Williams.

[4] D R Kuhn and T J Walsh and S Fries.
“Security Considerations for Voice Over IP Systems”.
National Institute of Standards and Technology.

[5] Defense Information Systems Agency.
“Voice Over Internet Protocol (VOIP) Security Technical Implementation Guide”.

[6] Juniper Networks.
“VoIP Security - best practices Outline”.

[7] H. Yan and K. Sripanidkulchai and H. Zhang and Z. Shae and D. Saha.
“Incorporating Active Fingerprinting into SPIT Prevention Systems”.
Third Annual VoIP Security Workshop (VSW’06).

[8] M. Chang and C. K. Poon.
"Catching the Picospams".
In International Syposium on Methodologies for Intelligent Systems (ISMIS 2005).

[9] J. Caballero and S. Venkataraman and P. Poosankam and M. G. Kang and D. Song and A. Blum.
"FiG: Automatic Fingerprint Generation".
The 14th Annual Network & Distributed System Security Conference (NDSS 2007).

[10] G. Shu and D. Lee.
"Network Protocol System Fingerprinting - A Formal Approach".
INFOCOM 2006. 25th IEEE International Conference on Computer Communications.

[11] G. Banks and M. Cova and V. Felmetsger and K. Almeroth and R. Kemmerer and G.Vigna.
"SNOOZE: Toward a Stateful NetwOrk prOtocol fuzZEr".
Springer, Lecture Notes in Computer Science 2006.

[12] J. Demott and R. Enbody and W .Punch
"Revolutionizing the Field of Grey-box Attack Surface Testing with Evolutionary Fuzzing"
Black Hat 2007

[13] R. Kaksonen.
"A Functional Method for Assessing Protocol. Implementation Security".
VTT Electronics. VTT Publications 448

[14] S. Embleton and S. Sparks and R. Cunningham.
“Sidewinder: An Evolutionary Guidance System for Malicious Input Crafting”.
Black Hat 2007

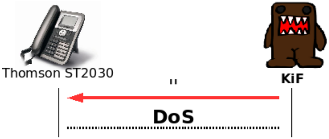
[15] M. Sutton and A. Greene and P. Amini.
“Fuzzing: Brute Force Vulnerability Discovery”.
Addison-Wesley Professional

[16] S. McAllister and E. Kirda and C. Krügel.
“Expanding human interactions for in-depth testing of web applications”.
RAID 2008, 11th Symposium on Recent Advances in Intrusion Detection

[17] I. Rouvellou and G. Hart.
“Inference of a probabilistic finite state machine from its output”.
IEEE Transactions on Systems, Man and Cybernetics 1995.

DoS: Basic Checking

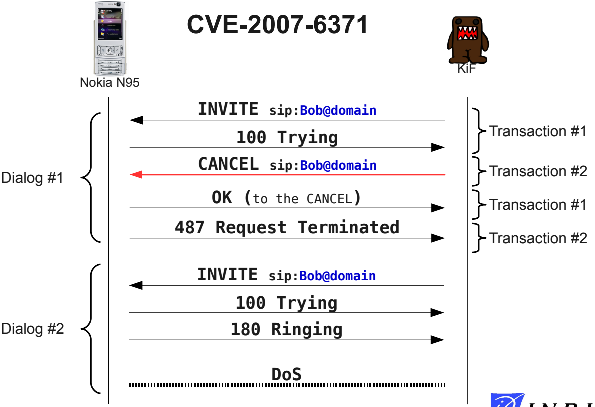
- When you have nothing to say ...
- One message can be sufficient to kill a phone
- A very simple message actually : empty UDP packet



Vulnerability by KiF
CVE-2007-4753

DoS: Stateful fuzzing

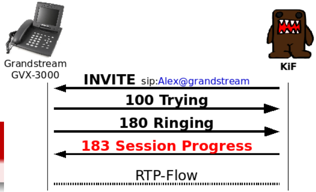
- One dialog prematurely cancelled
- Device reach a unstable state
- A new dialog hangs the device



Eavesdropping: Big Brother Dreams/Realities

- INVITE an entity but **... reply yourself**
- Remote entity accepts the call without asking
- Eavesdrops the conversation taken in the room

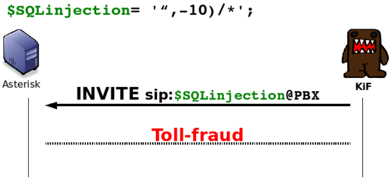
Requires stateful fuzzing to be performed !



Vulnerability by KiF
CVE-2007-4498

Injections: Why VoIP Insecurity is Really BAD?

- SQL Injections over SIP
  - SQL tables used for CDR
  - Unescaped inputs
  - Asterisk addons



$SQLinjection= '",-10)/*';

Asterisk

KiF

INVITE sip:$SQLinjection@PBX

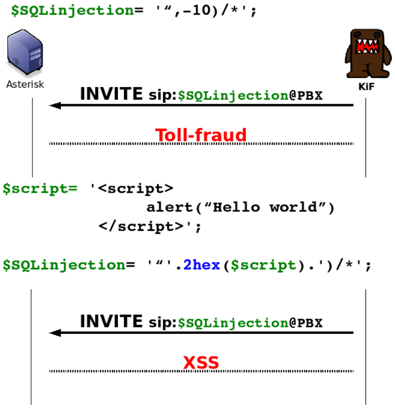
Toll-fraud

Vulnerability by KiF
CVE-2007-54881

INRIA MADYNES

Injections: Why VoIP Insecurity is Really BAD?

- SQL Injections over SIP
  - SQL tables used for CDR
  - Unescaped inputs
  - Asterisk addons
- Got one SQL injection?
  Have one XSS for free!
  - Unescaped database inputs
  - FreePBX, trixbox
- XSS via SQL injections
  through SIP



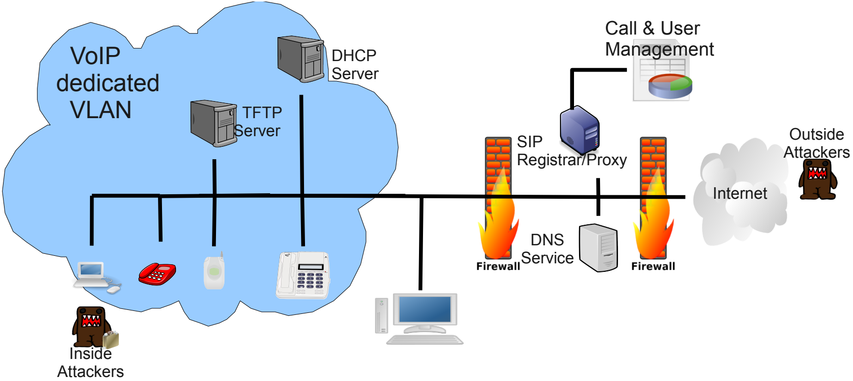
```
$SQLinjection= '“,-10)/*';
```

```
$script= '<script>
            alert(“Hello world”)
         </script>';

$SQLinjection= '“'.2hex($script).')/*';
```

Vulnerability by KiF
CVE-2007-54881

VoIP Deployment Layout [1, 2, 3, 4, 6, 5]


VoIP dedicated VLAN
DHCP Server
TFTP Server
Inside Attackers
Call & User Management
SIP Registrar/Proxy
Firewall
DNS Service
Firewall
Internet
Outside Attackers