# DENIAL OF SERVICE ATTACK AND PREVENTION ON SIP VOIP

Ge Zhang, Sven Ehlert, and Thomas Magedanz
Fraunhofer Institute FOKUS, Berlin, Germany

Presented by
Murad Kaplan

# OUTLINE

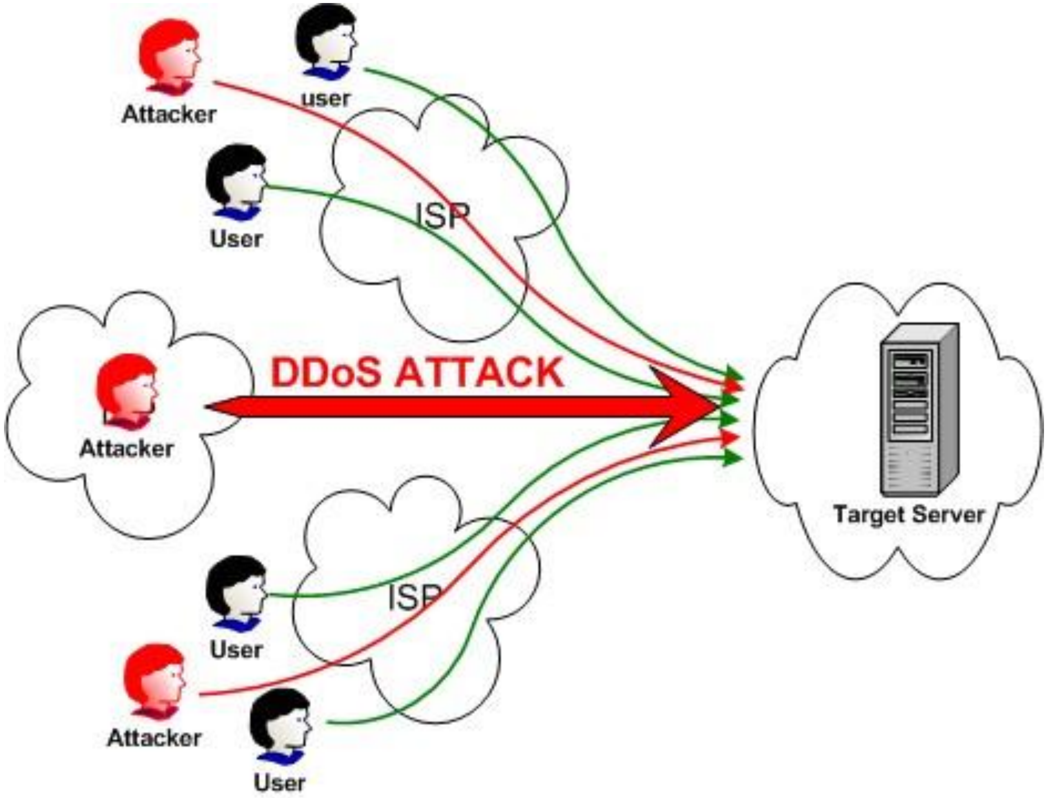
- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# INTRODUCTION

- DOS & DDOS
- Why DOS on VoIP ?
- Research questions

# DOS & DDOS

- Denial Of Service Attack



# WHY DOS ON VOIP ?

- Open environment.

- VoIP services are based on standardized and open technologies (i.e. SIP)

- Using servers reachable through the internet implemented in software and provided often over general purpose computing hardware.

# RESEARCH QUESTIONS:

- How to find a proper method to mitigate the effect of DoS attack via DNS request?

- Which factors of DNS cache and SIP proxy (e.g. caching replacement policy, cache entry number, parallel processes number of proxy, etc) are useful to deal with this problem?

- Which kind of combination of the useful factors is the most efficient?

# OUTLINE

- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# BACKGROUND

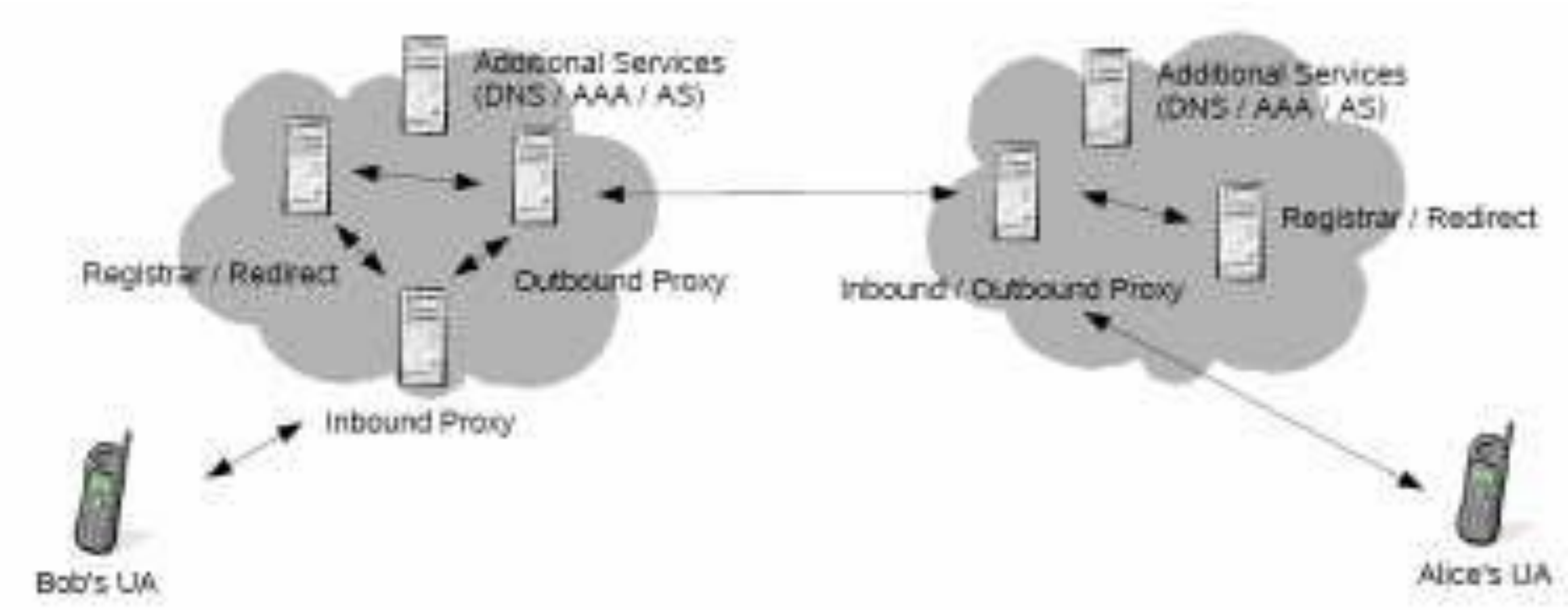
- Session Initial Protocol (SIP)
- Domain Name Server (DNS)
- DNS Usage in SIP Infrastructure

# SESSION INITIATION PROTOCOL (SIP)

- The standard for VoIP services in the Internet and next generation networks.
- Based protocol designed to establish or terminate a session between two parts.
- Also, SIP is the basic protocol of the next generation IP Multimedia Subsystem

# SIP INFRASTRUCTURE

- User Agent, Registrar, and Proxies

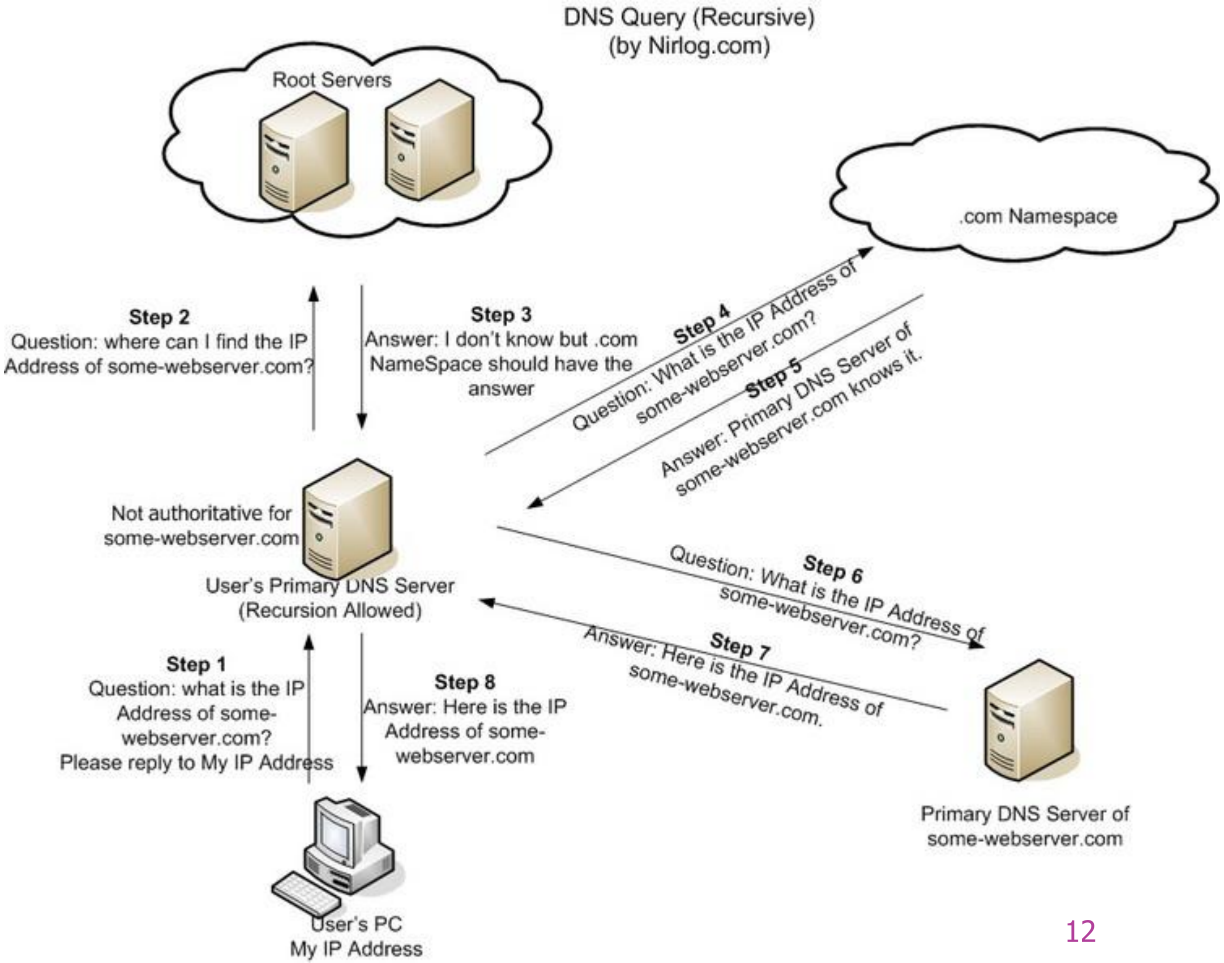


# DOMAIN NAME SERVER (DNS)

- The basis for most current internet services available today, including web and email.
- completely globally distributed and managed database.
- provides an essential service for Internet applications and users i.e. name resolution

# USER REQUESTS A DOMAIN RESOLVE

- The DNS server knows the name mapping.
- The DNS server does not know the name mapping.


DNS Query (Recursive)
(by Nirlog.com)
Root Servers
.com Namespace
Step 2
Question: where can I find the IP Address of some-webserver.com?
Step 3
Answer: I don't know but .com NameSpace should have the answer
Step 4
Question: What is the IP Address of some-webserver.com?
Step 5
Answer: Primary DNS Server of some-webserver.com knows it.
Not authoritative for some-webserver.com
User's Primary DNS Server
(Recursion Allowed)
Step 6
Question: What is the IP Address of some-webserver.com?
Step 7
Answer: Here is the IP Address of some-webserver.com.
Step 1
Question: what is the IP Address of some-webserver.com?
Please reply to My IP Address
Step 8
Answer: Here is the IP Address of some-webserver.com
Primary DNS Server of some-webserver.com
User's PC
My IP Address

# DNS USAGE IN SIP INFRASTRUCTURE

- Many of the header fields in a SIP message contain Fully Qualified Domain Names (FQDN) that need to be resolved for further processing from a SIP entity.

- To interconnect the Public Switched Telephone Network (PSTN) with a SIP network.

- SIP can utilize different transport layer protocols (e.g. UDP or TLS).

# OUTLINE

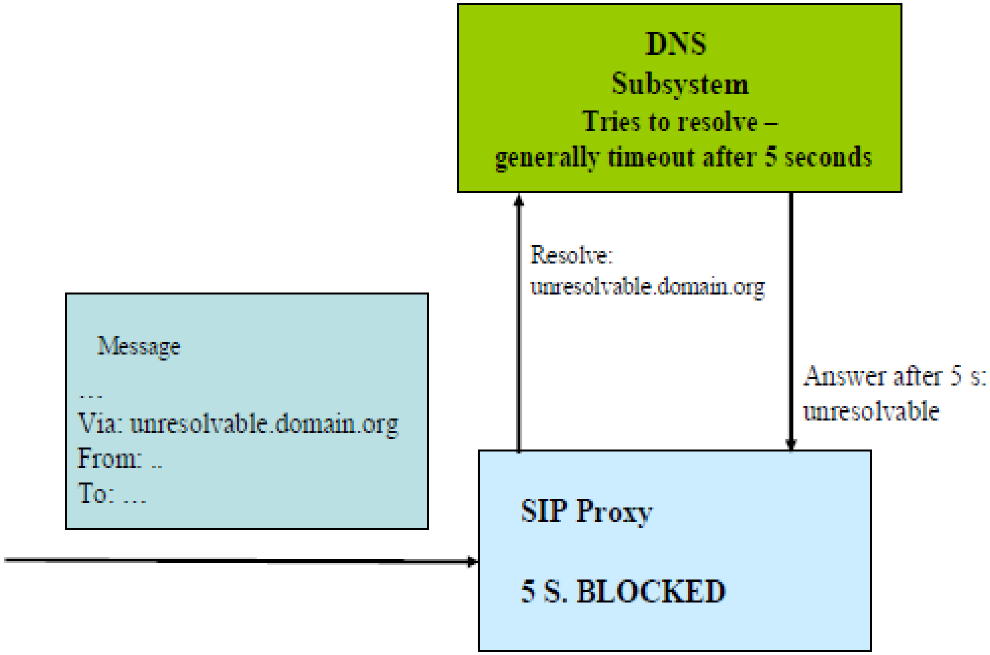
- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# SCOPE OF THE ATTACK

- The goal of a DoS attack is to render the service inoperable for as long as possible.
- Most effective against proxies or registrars/redirectors.
- The SIP DNS attack targets the DNS queries high processing time.

# SCOPE OF THE ATTACK

- Irresolvable domain name !!
- The attacker use URI that its mapping will not be in the cache of a name server and the URI will trigger a request to an authoritative name server that has a common low response time.



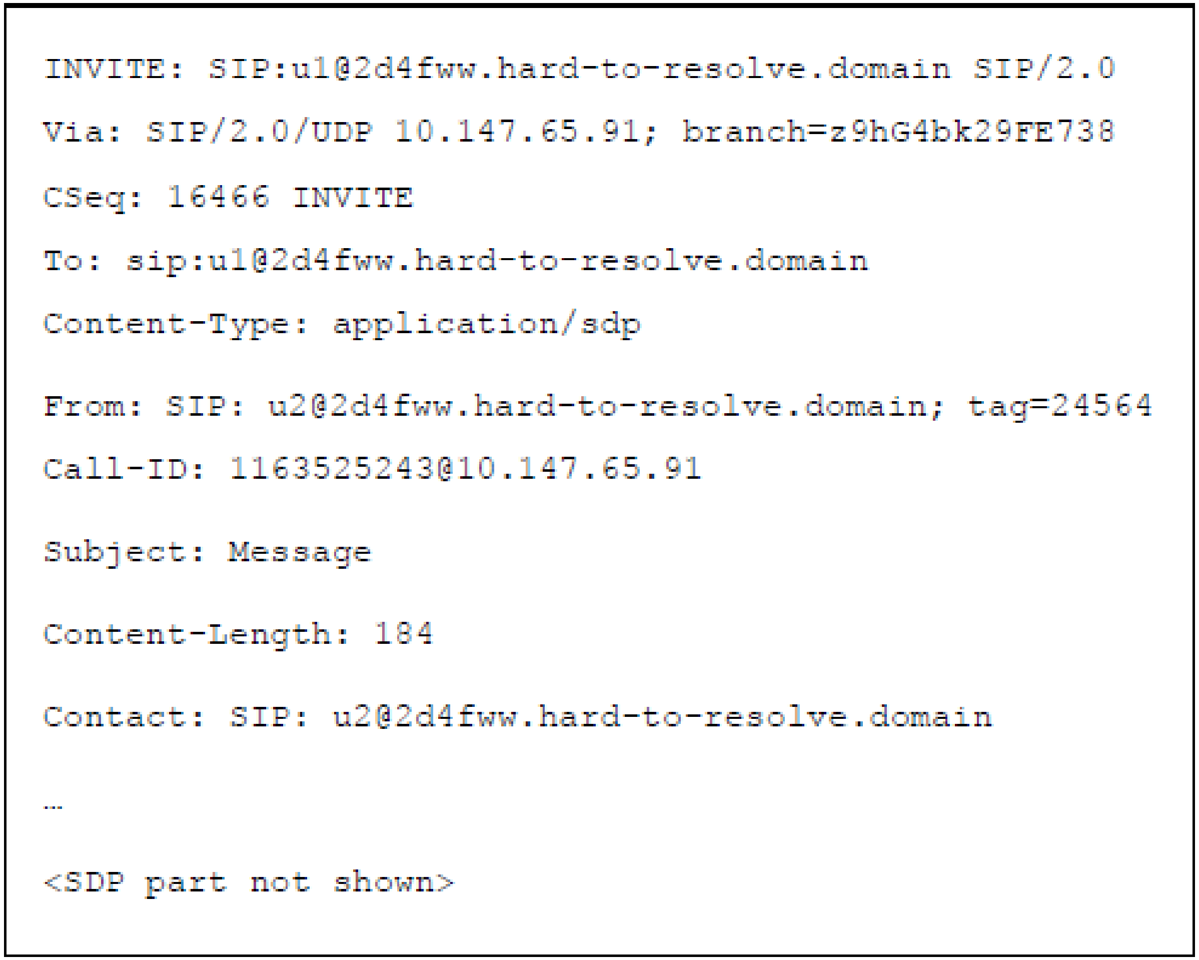
# SCOPE OF THE ATTACK

- Two ways to construct irresolvable domain name:

    1. Adding random host names to the left side of the address domain.
    2. Querying different name servers and measuring reply times.

*Such a message cannot easily be filtered out by a SIP server or an Intrusion Detection System.*

# SCOPE OF THE ATTACK



```
INVITE: SIP:u1@2d4fww.hard-to-resolve.domain SIP/2.0

Via: SIP/2.0/UDP 10.147.65.91; branch=z9hG4bk29FE738

CSeq: 16466 INVITE

To: sip:u1@2d4fww.hard-to-resolve.domain

Content-Type: application/sdp

From: SIP: u2@2d4fww.hard-to-resolve.domain; tag=24564

Call-ID: 1163525243@10.147.65.91

Subject: Message

Content-Length: 184

Contact: SIP: u2@2d4fww.hard-to-resolve.domain

…

<SDP part not shown>
```

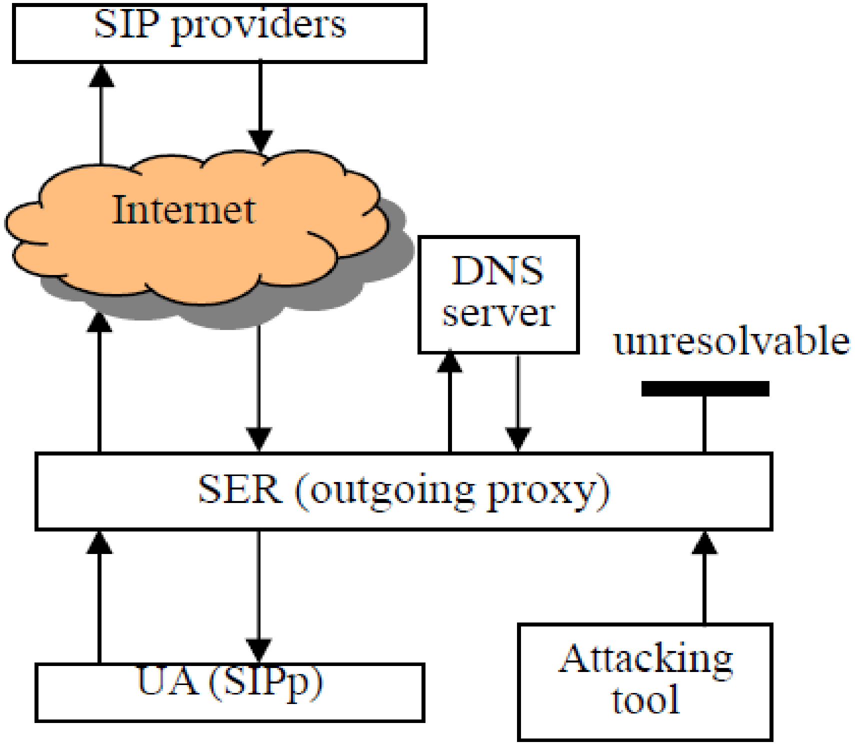
# OUTLINE

- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# TESTBED SETUP

Five Main component:

1. SIP proxy, using SIP Express (SER)
2. Local DNS server
3. Self implemented Attack Tool
4. User Agent (UA)
5. External SIP Provider



*The test bed was established on Pentium D double processor machines with 1 GB RAM (Proxy, User Agent, and Attack tool) running on Linux Operating Systems, equipped with 100 M Bps internet access.*

# TESTBED SETUP

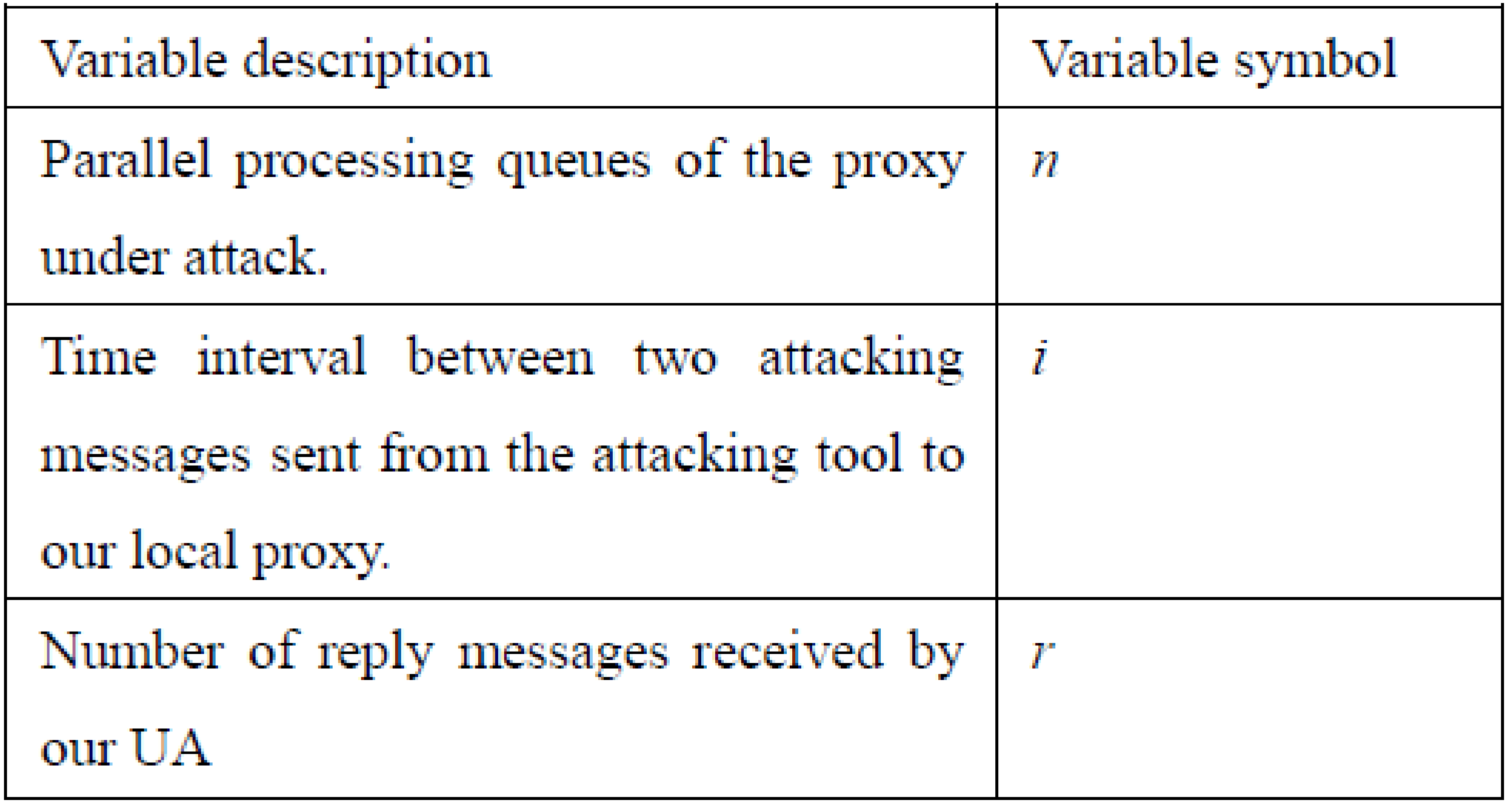
- The SIP proxy is setup first and can be configured to have different parallel processing queues n, with 2≤n≤64.
- According to SIP protocol, the UA has to REGISTER at a SIP server before it can INVITE others entities or receive INVITE messages. Therefore, REGISTER is the first essential step for the whole process and our experiments are focus on this step.
- UA configured to send continuously REGISTER messages from our local network to external SIP proxies. The external SIP register addressed are given to the UA in textual representation, as such our proxy has to resolve the domain before it can forward any request.
- The attacking tool is configured to send crafted messages containing hard resolvable domain names to the local outgoing SIP proxy. It is configurable by the attacking interval i seconds between two attacking messages.

# TESTBED SETUP

To measure the proxy performance, UA sends out 5000 register messages and counts the number of responses (r) that local proxy can process. If we can get any kind of response from a remote SIP server, it means the domain name of the server has successfully been resolved by our proxy.

*All measurements reported in the paper are the average value of 10 runs of the tests*

# EXPERIMENT VARIABLES



| Variable description | Variable symbol |
|---|---|
| Parallel processing queues of the proxy under attack. | $n$ |
| Time interval between two attacking messages sent from the attacking tool to our local proxy. | $i$ |
| Number of reply messages received by our UA | $r$ |

# OUTLINE

- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# DNS ATTACKS ON SIP PROXIES

- Two basic options for using DNS in a SIP proxy:

1. Synchronous DNS Implementation

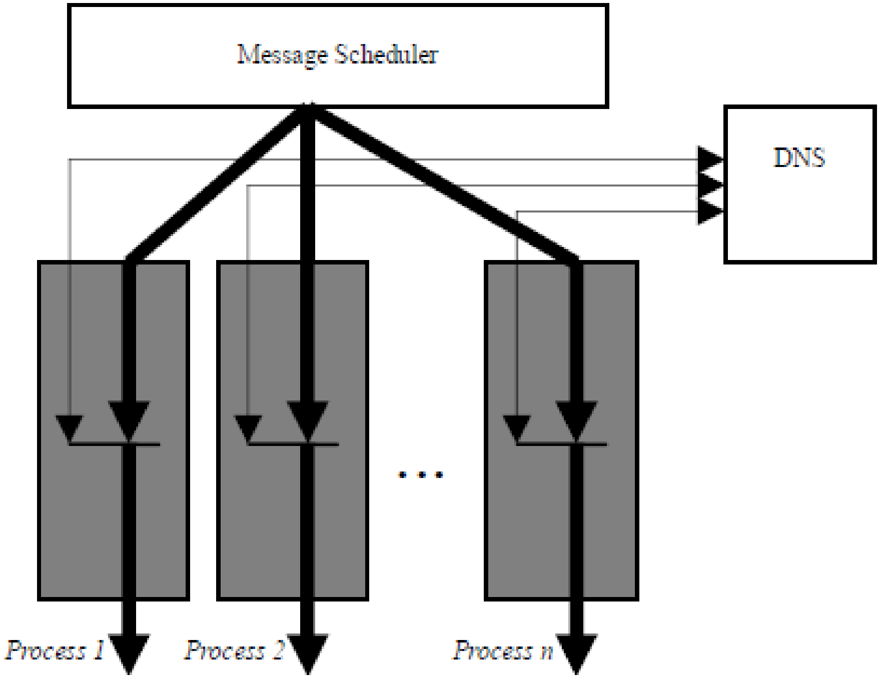
   SIP proxy sends a DNS request and waits for an answer, while waiting for the process that has issued the request the SIP proxy is blocked and unable to process new requests.

2. Asynchronous DNS Implementation

   SIP proxy issues DNS request continuously, saving state information after every request, once the reply to the DNS request arrives or timeout expires the proxy will be notified.

# DNS AND SYNCHRONOUS SIP PROXIES

Synchronous Scaling through Parallel Processing

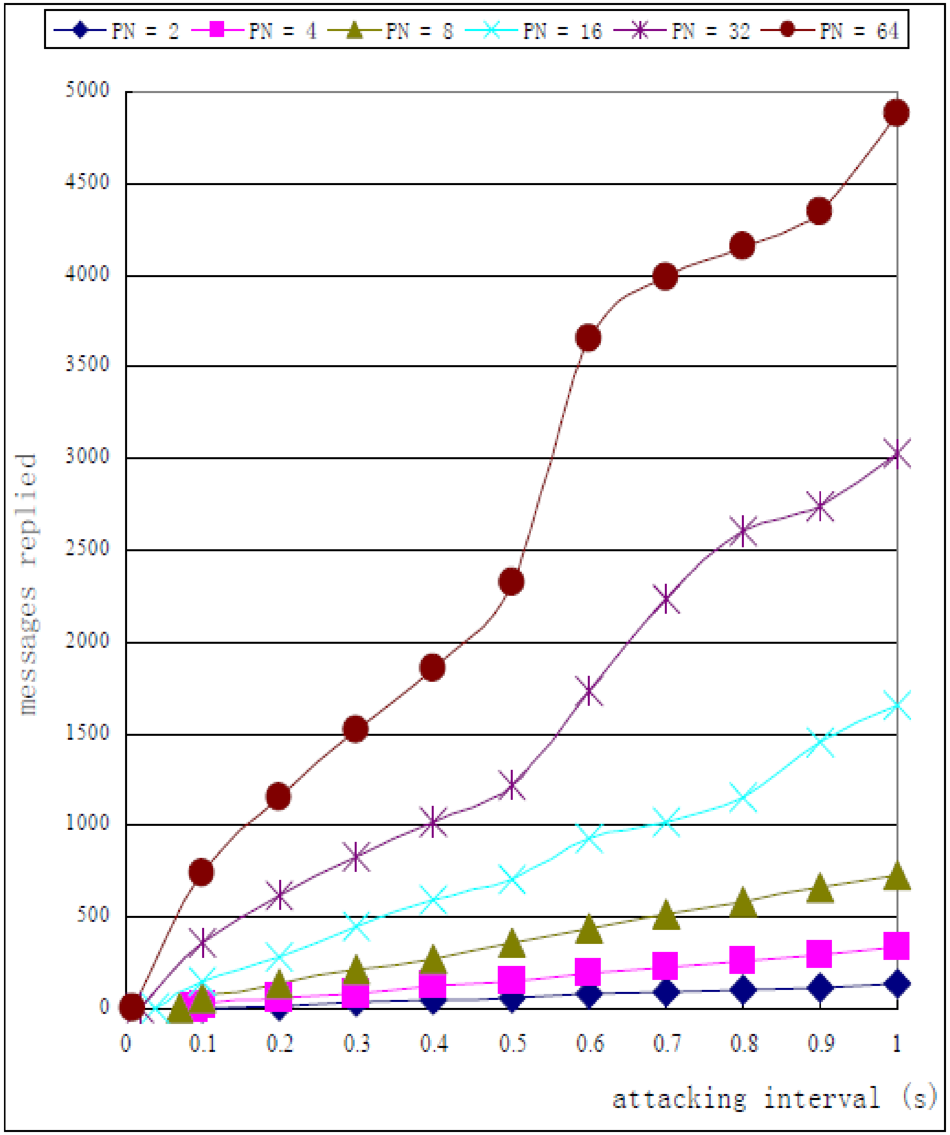


*Each process is responsible for parsing the message, initiating any DNS requests or requesting the execution of an application and finally forwarding the message.*

# DNS AND SYNCHRONOUS SIP PROXIES

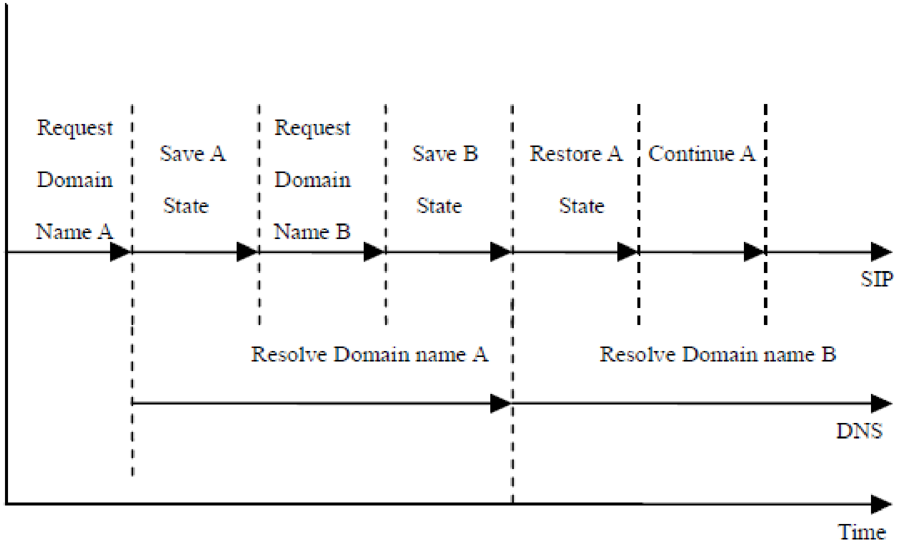
Processing Performance of the Local proxy for different processing queues *n* varying attacking intervals

- With few parallel processing queues (n ≤ 8) less than 20% of all potential messages can be processed.
- Decreasing the attacking interval down to 0.001 seconds (1000 attack message per second), even 64 parallel processing queues are completely starved.



# DNS AND ASYNCHRONOUS SIP PROXIES

- States of unfinished domain name resolving requests have been saved.
- The implementation complexity and memory requirements increase considerably.



- *SIP attack launched at a SIP proxy running on a machine with 8GB of RAM all memory can be depleted in about 30 seconds*

# OUTLINE

- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# NON-BLOCKING CACHE DESIGN

**DNS Attack Detection and Prevention (DADP)**

- SIP proxy is extended with a DNS cache.
- Blocking threshold, to insure that a SIP proxy continues function even under DNS attack.

$$S_q(t) = \begin{cases} 1, & \textit{a domain resolve call in} \\ & \textit{process queue } q \textit{ but not} \\ & \textit{returned at time } t, \\ 0, & \textit{otherwise} \end{cases}$$

SIP proxy *S* processing messages in synchronous manner, with *n* parallel processes.

# NON-BLOCKING CACHE DESIGN

- *H* indicates how many processes are concurrently resolving a domain name in time *t*.
- Proxy is blocked when *H* = *n*
- Minimum operation threshold *m*
- Whenever H ≥ R, where R = n – m, the proxy is informed that further DNS resolve request will have a high possibility to cause a DoS due to proxy blocking. As a consequence, the proxy will not try to resolve any domain names.
- Whenever H < R. Instead, the proxy assumes this address to be irresolvable, and continues its operation. As soon as H < R, the proxy will again perform DNS lookups.

$$H = \sum_{q=1}^{n} S_q(t)\ ,$$

# NON-BLOCKING CACHE DESIGN

**Implementation of DADP**

The extension of SER support:

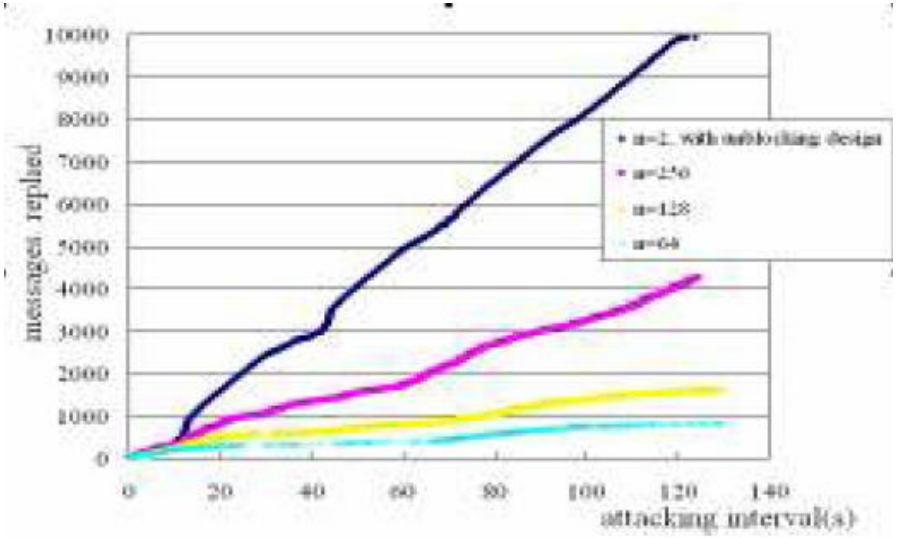
- Emergency process (whenever H ≥ n-1)
- Cashing both regular DNS entries (SRV, A records).
- Different cache replacement policies(FIFO, LRU, LFU, TC)

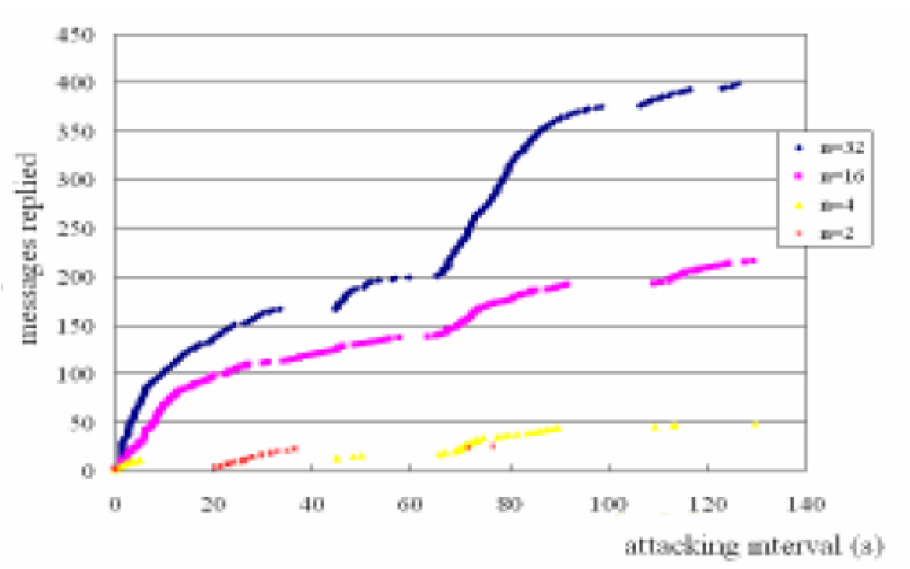
# NON-BLOCKING CACHE DESIGN

**Performance Evaluation of DADP**

- 10,000 SIP REGISTER messages were sent over a period of 140 seconds.
- These messages contained irresolvable domain names with 10 ms delay between each message

# NON-BLOCKING CACHE DESIGN





The more parallel queues *n* are configured, the more messages can be processed during an attack. With the DADP deployed nearly all messages can be processed independently of *n*.

# NON-BLOCKING CACHE DESIGN

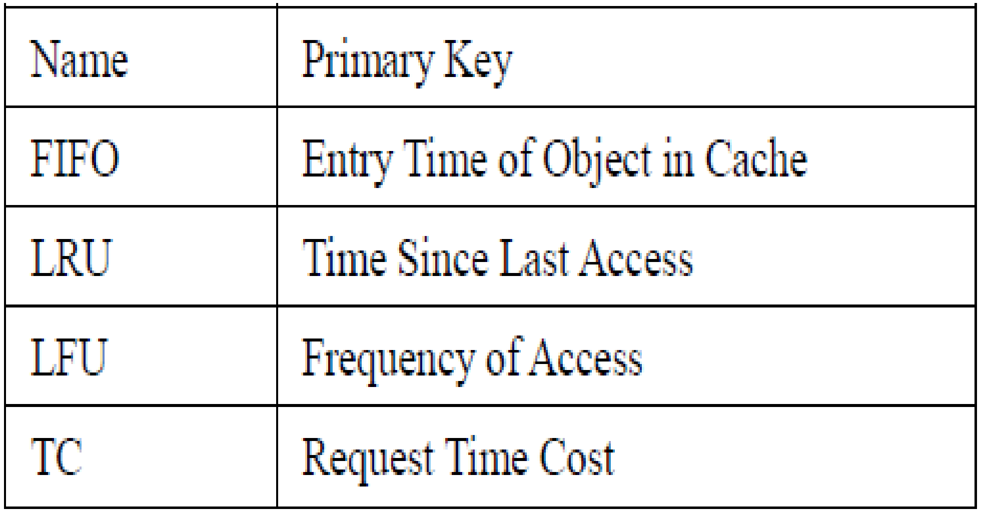
1- Cache Replacement Policies Evaluation

- Caching only successful requests.
- Once the cache is filled some existing cache entries will have to be deleted in order to make room for new items.
- Even a large cache entry storage can easily be allocated by an attacker with randomly generated invalid domain names.
- Considering four cache replacement policies (FIFO, LRU, LFU, TC)

# NON-BLOCKING CACHE DESIGN

Cache replacement policies

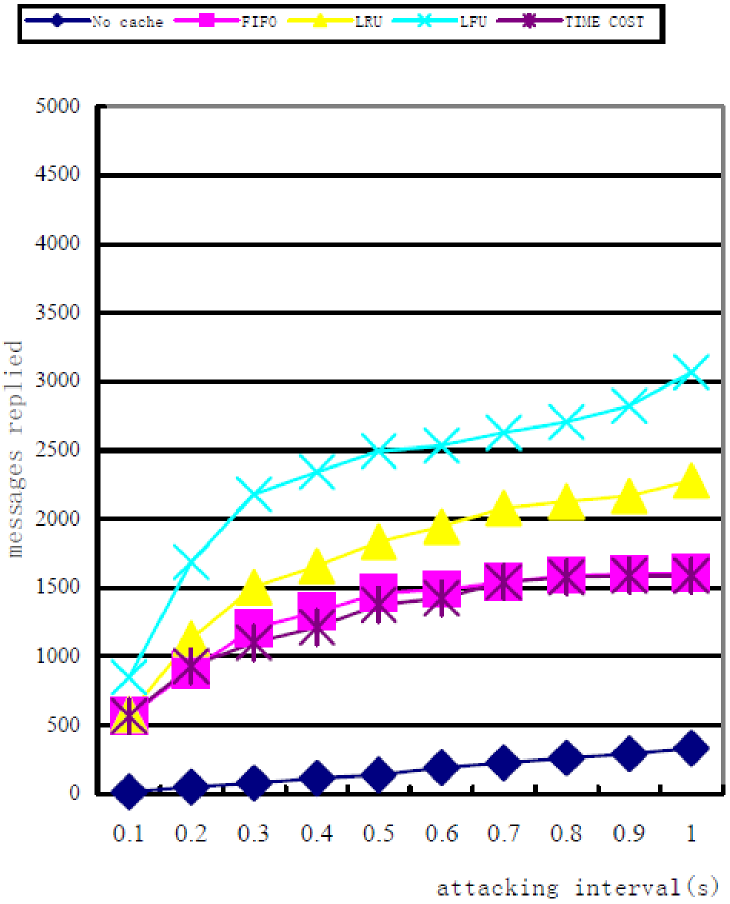
- First-in, First-out (FIFO).
- Least Recently Used (LRU).
- Least Frequently Used (LFU).
- Time Cost

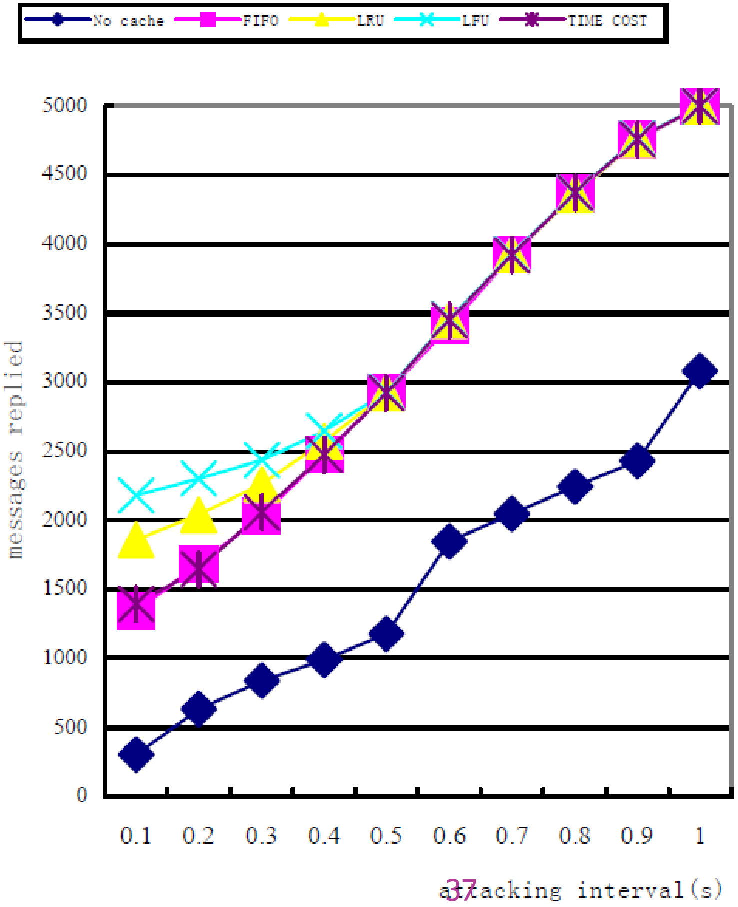


| Name | Primary Key |
|---|---|
| FIFO | Entry Time of Object in Cache |
| LRU | Time Since Last Access |
| LFU | Frequency of Access |
| TC | Request Time Cost |

# NON-BLOCKING CACHE DESIGN

repeated the experiment in with these four caching strategies twice, one time with *n*=4 parallel processing queues at the proxy and then again with *n*=32.





# NON-BLOCKING CACHE DESIGN

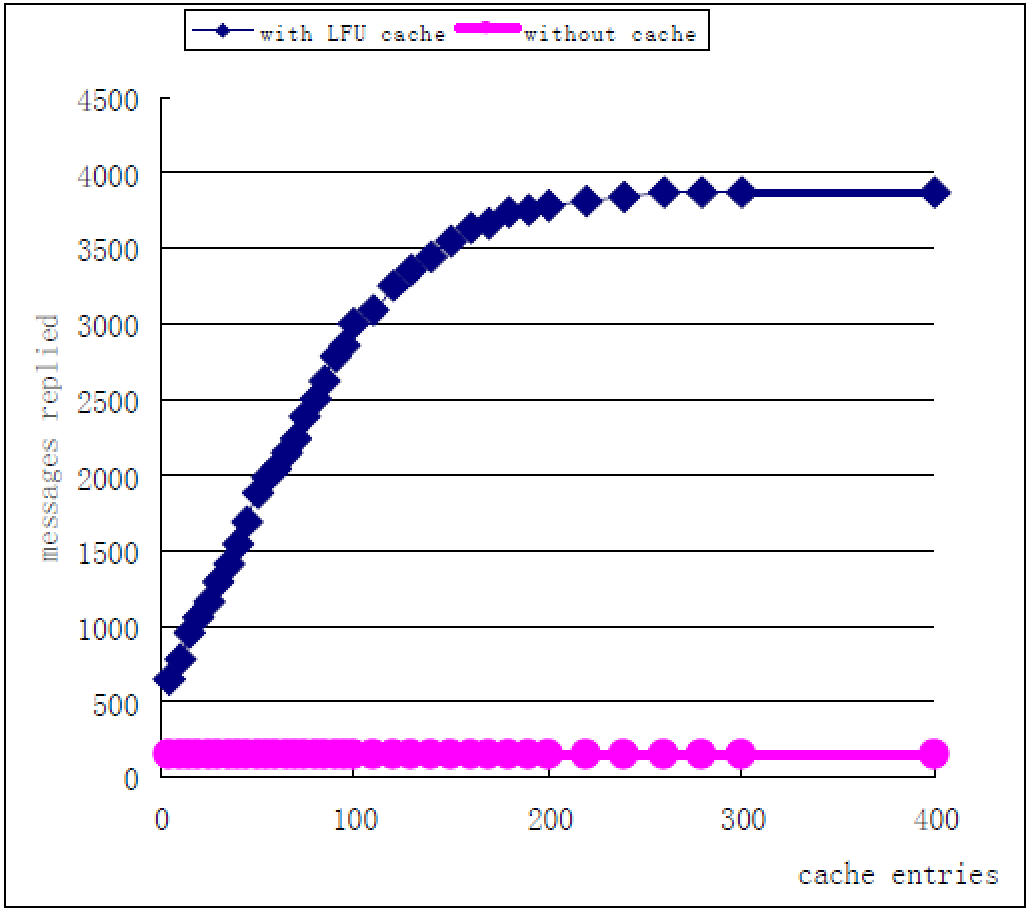
## 2- Evaluation of Cache Entry Numbers

- The relationship of caching performance in relation to caching entries *e*.
- The number of cache entries is limited while the amount of the domain names in the real world is almost unlimited.
- More cache entries will cost more memory and CPU resources to support them.

# NON-BLOCKING CACHE DESIGN

set *n*=4 and *i*=0.5

- Even with only 5 entries in the cache, the proxy still processes at least twice as much as without DADP.
- More cache entries substantially increase the performance of the proxy.



# OUTLINE

- Introduction
- Background
- Scope of The Attack
- Testbed Setup
- DNS Attacks on SIP Proxies
- Non-Blocking Cache Design
- Conclusion and Future Work

# CONCLUSION AND FUTURE WORK

- Denial of Service attacks can limit SIP server operation to a large amount.
- In synchronous manner, threshold can be the percentage of blocked process in SIP server.
- In asynchronous manner, threshold could represent the percentage of used memory.
- Use LFU, in case the used cache can not accommodate all possible DNS names.
- In the future, planning how to the DADP solutions works when the attack combined with another common DNS related attack, called DNS cache poisoning 0

# NOTES

- Many grammar mistakes.
- All of the figures need to be named again.

Thanks !

Murad Kaplan