“How I DOS’ed My Bank”

(DTMF Input Processing – Easy DOS Attacks via Phone Lines)

60 Min, Genres: Action, Comedy, Sci-Fi, Thriller, Family

Our paper would be on DTMF input processing algorithms [DSP] , that are often embed into PBX, IVR, Telephone routers and other devices that process DTMF input. PBX and IVR servers are often deployed for running Phone Banking App Servers, Call Center Application and other systems that uses phone to interact with them. If an attacker could trigger exception in DTMF processing algorithms, then they could crash the entire application server making a single phone call, causing the entire Phone banking in accessible, or no calls to the costumer service goes through. One such denial of Service could cause a lot of panic and the amount of damage would be pretty huge. We will be demonstrating lot of amusing remote DTMF attacks on Phone Banking, Tele-Voting, and Customer Support applications using DTMF. This talk is recommended for Pentesters, PCI|DSS consultants, Telephone Companies, Banks or anyone who uses a device interacted via Telephone.

Creation: Rahul Sasi (fb1h2s)
Production: Garage4Hackers
Distribution: NullCon
Cast: Me, Bank X, Vulnerable
Codes, IVR, PBX, DTMF
Guest Roles: Java, .Net codes.

Story Line (Introduction):

Input validation attacks and memory corruption attacks are common, and the criticality of finding a DOS attack on a service like HTTP is consider a lot critical considering the attack surface and easiness of attack. Even if we could trigger an exception in an Apache Web server and crash them, that would be a huge loss for corporates and individuals hosting critical applications on these systems.

Our paper would be on DTMF input processing algorithms [DSP], that are often embed into PBX, IVR, Telephone routers and other devices that process DTMF input. PBX and IVR servers are often deployed for running Phone Banking App Servers, Call Center Application and other systems that uses phone to interact with them. If an attacker could trigger exception in DTMF processing algorithms, then they could crash the entire application server making a single phone call, causing the entire Phone banking in accessible, or no calls to the costumer service goes through. One such denial of Service could cause a lot of panic and the amount of damage would be pretty huge. We will be demonstrating lot of amusing remote DTMF attacks on Phone Banking, Tele-Voting, and Customer Support applications using DTMF. This talk is recommended for Pen-testers, PCI|DSS consultants, Telephone Companies, Banks or anyone who uses a device interacted via Telephone. Attending this talk would make you aware of these attacks and preventive measures before the bad guys find it. It also helps u to handle proposals for IVR [Tele App] audits and help identifying the potential security problems.

History Of the Research:

We have already demonstrated at [BlackHat EU] how it was possible to remotely attack IVR implementations [Phone Banking], since IVR uses DTMF as input and Voice as output, and if an attacker could trigger errors in internal server using custom DTMF payloads, the IVR systems would read out the error to the attacker. This way it was easy to finger print Internal Network and extract critical information out of them. We also demonstrated input validation attacks on Phone Banking VXML [voice-xml] implementation of IVR applications.

Telephone based networks like IVR, PBX all are considered secure since they are not connected to Internet and they use a Peer to Peer GSM, CDMA, Wired Telephone line for transactions. And also the limitations of Payloads that could be constructed out of DTMF 0-9 *# ABCD keys are less. And this is one reason there is no Firewall engine to monitor Telephone traffic. In our current paper we would be digging deep and would explain vulnerabilities in implementations of DTMF detection algorithms and Libraries in [C, Java, .Net] platforms that are used in DTMF detection and are embedded into IVR, PBX and Telephone routing engines and many other products. Bugs in these embedded algorithms that takes

DTMF voice as input, could be easily triggered via a phone call. Vulnerabilities in implementation of DTMF detection algorithms would be the highlight of our talk.

In Short:

Previous Research Paper Contents:

- IVR[Interactive Voice Response System] are used for information retrieval and Remote Management via Telephone lines.
- We demonstrated how it was possible to remotely attack IVR implementations [Phone Banking].
- IVR uses Voice Commands, DTMF as input and Voice as output.
- If an attacker could trigger errors in internal server using custom DTMF|Voice payloads, the IVR systems would read out the error to the attacker.
- It was easy to finger print Internal Network and extract critical information out of them.
- We demonstrated input validation attacks on Phone Banking VXML [voice-xml] implementation of IVR applications.
- We explained how Buffer Overflows and SQL injections and Bruteforce attacks could be conducted on IVR based Applications.

Current Research Paper Contents:

- Previously DTMF detection was done using Hardware devices, currently it's replaced by software modules using DSP [Discrete Fourier Transform] algorithms.
- If an attacker could trigger exception in DTMF processing algorithms, then they could crash the entire application server making a single phone call.
- Goertzel's algorithm is mostly preferred for DTMF detection algorithm.
- Vulnerabilities in implementation of DTMF detection algorithms.
- Goertzel's algorithm explained in details with implementation in [C,Java]
- In Goertzel's algorithm input, the wavelength is a user-controlled value.
  - How Goertzel's algorithm computes the right DTMF tones using DFT.

    How the Sin|Cos table are computed, the CPU instructions and all for DFT
- How algorithm implementation could be used to trigger exceptions | memory exhaustion.
- Building a DTMF fuzzer and testing DTMF processing Codes|Scripts.

- Crashing appliances that have got DTMF input with our Fuzzer.

The following things would be our current paper.

Technical Details (Abstract):

DTMF decoders in the past used special IC that accepts DTMF waveforms and produced the 16 possible outputs 0-9 *# ABCD. But currently these chips | devices are no longer used and are replaced with software Implementation of these using DSP (Digital Signal Processing DFT) Discrete Fourier Transform. In which the algorithm that is mostly preferred for DTMF detection is the Goertzel's algorithm and there are few others implementations too. These implementation of DTMF detection are embedded into IVR, PBX, Telephone Routers, and any place where DTMF is used. The applications of IVR, PBX are often seen in various places like Phone Banking, Tele-Voting, Voice Mail, Customer Support, Automobile hands free operation etc.

DTMF:

DTMF signals are generated from a combination of two sinusoidal Low band and High Band frequency to transmit 16 digits, A-D , 0-9, and * # tones. And on the other end there would be a detector to convert the transmitted DTMF tones to data.

DTMF Decoding Algorithm:

Goertzel algorithm is one among the mostly used DTMF detection algorithms.

The Goertzel algorithm computes a sequence using DFT, 16 samples of DFT are computed for the 16 tones.

$$s(n) = x(n) + 2\cos(2\pi\omega)s(n-1) - s(n-2)$$

Or

$$Q_n = x(n) + 2\cos\left(\frac{2\pi k}{N}\right)Q_{n-1} - Q_{n-2}$$

- Here ω is the wavelength of interest

Therefor:

$$|y_k(N)| = Q^2(N) + Q^2(N-1) - 2\cos\left(\frac{2\pi k}{N}\right)Q(N)Q(N-1)$$

And the value of k determines the tone we are trying to determine.

$$k = N \times \frac{f_{tone}}{f_s}$$

f_{tone} = Frequency of tone
f_s = Sampling Frequency

A Pseudo code for the implementation would be as follows.

```
standard_frequency = output_frequency / sample_rate;
coeff = 2*cos(2*PI*standard_frequency);
for each sample, x[n],
  s = x[n] + coeff*s_prev - s_prev2;
  s_prev2 = s_prev;
  s_prev = s;
end
power = s_prev2*s_prev2 + s_prev*s_prev -
coeff*s_prev*s_prev2
```

What we control [Fuzzing]:

In the above we control the Frequency and Time duration of the tone, and this would be the value we would be fuzzing on.

The Attacks [The Villan] :

The inputs to the algorithm are Frequency and Time duration of tone at times.

We saw many faulty implementations that could trigger, Memory Exhaustion, Integer Overflows (local dos), Un handled exceptions while fuzzing with the frequency and time variables.

Integer Overflows: [Local Denial of Service]:

We would be demonstrating vulnerable code that are prone to overflow, Integer overflows and stack corruption, which could be triggered over a DTMF tone | phone call. Even though these vulnerabilities would be hard | impossible for code execution, they would be enough to cause a DOS to the embedded services having these DTFM detection algorithms.

Memory exhaustion:

Even though DFT uses a limited set of Sine Cos tables and use limited CPU, it was possible to overload the CPU using few unique payloads and that created an overloading Sine Cos tables. . This way it would be possible even to crash| disable the Application server itself. We will demonstrate a Phone banking scenario.

Information Disclosure:

We would be demonstrating, in what way we would be able to extract sensitive information about the application's hosted environment with these sorts of bugs. Since application that use DTMF algorithms are mainly phone based application, most of the case it was possible to extract output in form of audio data.

Overall:

We would be explaining how many phone based protocols an attacker could disrupt and the inability of current security precautions to stop them. We will have lot of live demos demonstrating the paper and some deep technical talk on the attacks.

There is no other paper | talks documenting on Phone Banking Security, DTMF Fuzzing and IVR app security, nor there are any standards, framework, which explain the secure building of these systems nor methodologies to attack them. Since phone banking and Costumer support services are a lot popular these days. And since the no of costumers who prefer Phone based services are huge, and attacks are fairly easy to perform over a Phone call the talk would be of high interest. It would be having high technical contents and amusing demos that would interest a Technical person as well as a managerial post.