

## আপনার প্রোগ্রামকে পাইরেসি হতে রক্ষা করবেন কিভাবে ?

# সফটওয়্যার পাইরেসি প্রতিরোধের কৌশল

**প্রতিনিয়তই** যেসব প্রোগ্রাম বা সফটওয়্যারসমূহ ব্যবহার করছি তার কয়টি আমাদের কেনা ? কী বললেন ? বাজার থেকে সিডি কিনে এনেছেন ? বস্তুত আমরা প্রায় কেউই কিনে ব্যবহার করছি না এসব প্রোগ্রাম, যা ব্যবহার করছি তা পাইরেটেড কপি। আমরা সিডিতে যা কিনছি সবই চুরি করা। আমাদের যখন আসল প্রোগ্রাম না কিনলেও হচ্ছে, তখন সমস্যাটা কোনখানে ?

সমস্যা সেখানেই যখন আপনি একজন প্রোগ্রামার বা আপনার সফটওয়্যার প্রতিষ্ঠানটি মাইক্রোসফটের মতো কোন জায়গায় নয়। আমরা এখানে আজ আপনাকে দেখানোর চেষ্টা করবে কিছু কার্যকরী কৌশল যার প্রয়োগে আপনার প্রোগ্রামটি চুরি হয়ে যাওয়া থেকে রক্ষা করতে পারেন।

আসুন জেনে নেয়া যাক, বিভিন্ন ধরনের প্রোগ্রামের জন্য কী কী ধরনের ব্যাপারগুলো আপনাকে মোকাবেলা করতে হতে পারে একজন প্রোগ্রামার বা সফটওয়্যার ডেভেলপার হিসেবে। এখানে আমরা আরো দেখব কী করে একজন ক্রয়কার একটি প্রোগ্রামের প্রোটেকশন ভাঙে সেইসাথে দেখব তা রোধ করার উপায়সমূহ।

### শেয়ারওয়্যার প্রোগ্রামের

#### ক্ষেত্রে

শেয়ারওয়্যার প্রোগ্রামের কপি প্রোটেকশনের ক্ষেত্রে দুটি বড় সমস্যা রয়েছে। প্রথমটি হলো— কীভাবে একজন ক্রয়কারকে আপনার

শেয়ারওয়্যারটিকে সম্পূর্ণ ব্যবহারযোগ্য ভাসনে রূপান্তর করা থেকে ঠেকাবেন। দ্বিতীয়টি হলো পূর্ণাঙ্গ একটি সফটওয়্যারেরই কপি প্রোটেকশন, মানে হচ্ছে আপনি কীভাবে নিশ্চিত করবেন যে, আপনার তৈরি প্রোগ্রামটি কোনো বৈধ ক্রেতাই ব্যবহার করছে। আসুন আগে জেনে নেই কী কী ভাবে শেয়ারওয়্যার প্রোগ্রাম তৈরি করা হয়। একটি প্রোগ্রামের শেয়ার ভাসন টার্গেট ব্যবহারকারীকে দেয়ার পেছনে একজন প্রোগ্রাম রচয়িতার মূলত দুটি উদ্দেশ্য কাজ করে থাকে। একদিকে যেমন তিনি তার প্রোগ্রামের সকল অপশন, ইউজ্যাবিলিটি, সুবিধা সম্পূর্ণরূপে ভুলে ধরে ক্রেতা ব্যবহারকারীকে আকৃষ্ট করতে চান। অপরদিকে তিনি অবশ্যই চান প্রোগ্রামটি কেবল টাকার বিনিময়ে কিনে সবাই ব্যবহার করুক। নিচে



কতগুলো বহুল ব্যবহৃত শেয়ারওয়্যার বা ডেমো প্রোগ্রাম তৈরির পন্থা আলোচনা করা হলো—

### ডেমো প্রোগ্রাম

একজন সম্ভাব্য ক্রেতা একটি অপেক্ষাকৃত ছোট ভাসনে প্রোগ্রামটি পাবে, এর মধ্যে কিছু কিছু অপশন সম্পূর্ণ অনুপস্থিত থাকবে। আর এটি ব্যবহার করে একজন এটি কিনবেও।

এ ধরনের ডেমো প্রোগ্রামকে কিন্তু কোনোভাবেই সম্পূর্ণ প্রোগ্রামে রূপান্তরের কোনো উপায় থাকে না। এখন চলুন দুটি উদাহরণ দেখি— ধরা যাক একটি টেলিফোন ডিরেক্টরির ডেমো ভাসনের শুধু একটি ফোন স্টেশনে ফোন নম্বরই রাখা আছে। অন্য একটি উদাহরণ হতে পারে একটি গ্রাফিক্স এডিটরের, সেটি আপনাকে একটি পূর্বনির্ধারিত সাইজের চেয়ে বড় ছবি নিয়ে কাজ করতে দেবে না। এটা খুবই সম্ভব যে, কোনো চালাক হ্যাকার এই গ্রাফিক্স প্রোগ্রামটির এই সীমাবদ্ধতা দূর করে দিতে পারে। কিন্তু প্রথম প্রোগ্রামটির ক্ষেত্রে ফোন ডিরেক্টরির একটি অংশ একেবারেই নেই, আর তাই এটি হ্যাক করাও সম্ভব নয়।

### ইনকমপ্লিট ফাংশনালিটি

একটি শেয়ারওয়্যার ভাসনে অনেক সময় অনেক দরকারি সুবিধা দেয়া থাকে না। যা পূর্ণাঙ্গ প্রোগ্রামে থাকবে সেটি রেজিস্ট্রেশন-এর পর। তাই কোন কোন ফাংশনগুলো ট্রায়াল পিরিয়ডের সময়



অচল বা অকেজো করা থাকবে তা নির্ধারণ করাও একটি গুরুত্বপূর্ণ ব্যাপার। কেননা বাজারে আরো অন্যান্য প্রোগ্রামের তুলনায় আপনারটি লোভনীয় হতে হবে। আপনার কোনো ভুলের জন্য আপনার প্রতিদ্বন্দ্বী প্রোগ্রামটিই একজন ব্যবহারকারী কেনার সিদ্ধান্ত নিতে পারে। তাই অনেক বেশি পরিমাণে ফাংশনটি যদি অফ করে ডেমো তৈরি

করেন তাহলে হয়তো একজন ক্রেতা সেটি আর ব্যবহার করার চিন্তা করবে না সুতরাং ওটি কেনার জন্য তা মনে উপযোগও সম্ভব হবে না। এক্ষেত্রে First Impression lasts longer প্রবাদটি কাজ করে থাকে। তাই এটাও একটি গুরুত্বপূর্ণ ব্যাপার যে, আপনার ডেমো প্রোগ্রামটি একজন ক্রেতাকে যতটা সম্ভব প্ররোচিত করবে এটি কেনার জন্য। তাই সে ফাংশনালিটিগুলো যেন এমন না হয় যে প্রথমই উৎকট এবং দৃষ্টিকটুভাবে ব্যবহারকারীর নজর কাড়ে। আবার এ ফাংশনগুলো আকর্ষণীয় এবং দরকারিও হওয়া দরকার। একই সঙ্গে ক্রেতার চোখে এটি পরিষ্কার করতে হবে যে, প্রোগ্রামটি রেজিস্টার করলে কতটা বেশি সুবিধা সে পাবে আর আসলেই সে ফাংশনগুলো কতটা কার্যকর হবে। অর্থাৎ তার বিশ্বাস ও আস্থা অর্জন করতে হবে। ট্রায়াল প্রোগ্রামটিতে ইউজার যেন অভ্যস্ত হয়ে ওঠে এবং এখনি এর ফুল ভাসনটা হাতে চাই এরকম স্পৃহা যেন জন্মায়। সত্যিই কঠিন কাজ বৈকি!

### ব্যবহারের দিন তারিখ ও

### ব্যবহার সংখ্যা সংক্রান্ত

### সীমাবদ্ধতা অর্পন

এটিও একটি জনপ্রিয় উপায়। আগে বর্ণিত মেথডগুলোর অসুবিধাগুলো এটিতে নেই— ট্রায়াল দেয়ার সময়টিতে ব্যবহারকারী এর সবরকম সুবিধাই উপভোগ করতে পারবে। অনেকেই এই মেথডে তাদের বেটা ভাসন সফটওয়্যার বিতরণ করে থাকেন ব্যবহারের ওপর সীমা আরোপের জন্য। এমনকি মাইক্রোসফটও কিছু কিছু ক্ষেত্রে করে থাকে।

### বিরক্তিকর আচরণ

ট্রায়াল ব্যবহারের সময়কালীন এ জাতীয় প্রোগ্রামগুলো মাঝে মাঝে ব্যবহারকারীকে জ্বালায় কিন্তু প্রোগ্রামটির সকল ফিচারই ব্যবহার করতে দেয়। সাধারণত এটি করা হয় প্রোগ্রামটি রেজিস্টার করে ব্যবহার করার কথা মনে করিয়ে দিতে। একটি ভালো উদাহরণ হতে পারে জনপ্রিয় উইনজিপ প্রোগ্রামটি, যেটি প্রত্যেকবার রান করার সময় রেজিস্ট্রেশনের রিমাইন্ডার দেয়। এ ধরনের মেথড প্রথমগুলো থেকে আলাদা একারণে যে, এতে রেজিস্টার্ড ভাসনে



কোনো কিছু সংযুক্ত করতে হয় না (প্রোগ্রামটি পূর্ণাঙ্গ ফিচারসমৃদ্ধ করতে) আবার তৃতীয় পক্ষের ন্যায় কোনো কিছু রিমুভ করারও প্রয়োজন পড়ে না। তাহলে দেখা যাচ্ছে তৃতীয় পক্ষের নিউজ নির্ভরযোগ্যতার দিক দিয়ে অধিক গ্রহণযোগ্য।

## রেজিস্ট্রেশন ইনফো

এই অপশনটি রাখা হয় সাধুসত্ত্ব ব্যক্তিদের জন্য। এক্ষেত্রে প্রোগ্রামটি কেবল জানাবে যে এটি এখনো রেজিস্টার্ড নয় বা কার জন্য রেজিস্টার্ড (স্ক্রিনে দেখাতে পারে অথবা About বক্সে)। এ ধরনের পদ্ধতি প্রায় সব প্রোগ্রামেই থাকে কিন্তু এটিই এককভাবে থাকে না। কেন না সাধু সত্ত্ব ব্যক্তি তো আর ঘরে ঘরে বাস করে না!

## আপগ্রেড

এ পদ্ধতিতে নতুন ভার্সনের প্রোগ্রামটি কেবল তখন ইনস্টল করতে দেবে যখন ব্যবহারকারীর পিসিতে আগের কোনো রেজিস্টার্ড ভার্সন আছে। সাধারণত এটি একটি অপূর্ণাঙ্গ প্রোগ্রাম যা আগের ভার্সনের Reusable অংশগুলোও কাজে লাগায় প্রয়োজনমতো ক্ষেত্রে। তবে এর ব্যতায়ও ঘটে যেমন— Adobe Photoshop বা Illustrator-এর কয়েকটি আপগ্রেড ভার্সনে আসলে ফুলকপিই দেয়া হয়েছে, আগের কোনো অংশের পূর্বব্যবহার করে না।

## প্রোগ্রাম রেজিস্ট্রেশন

কীভাবে একটি প্রোগ্রাম রেজিস্ট্রেশন হয়ে থাকে? এমন হতে পারে যে, প্রোগ্রামটির টাকা পরিশোধ করার পর, ব্যবহারকারী পার্সেল-এর মাধ্যমে সম্পূর্ণ নতুন (কমপ্লিট) ভার্সনটি একটি চকচকে বক্সে প্যাকেটকৃত অবস্থায় পাবে যার সাথে পূর্ণাঙ্গ ডকুমেন্টেশন ইত্যাদি দেয়া হবে। এ পদ্ধতির সমস্যা হলো ব্যবহারকারী সঙ্গে সঙ্গেই ফলাফল পাচ্ছে না অর্থাৎ অর্থ লেনদেনের সাথে সাথেই প্রোগ্রামটি হাতে পাচ্ছে না (খরা যাক লেনদেনটি হয়েছিল অনলাইনে ক্রেডিটকার্ডেও সেক্ষেত্রে তাকে অপেক্ষা করতে হবে কম করে হলেও বেশ কয়েকদিন। আর যদি এর ডাউনলোড ভার্সন ইন্টারনেটেই বিতরণ করা হয় তাহলে চকচকে বক্স, ফ্লপি/সিডি, বই/ম্যানুয়াল খরচ এবং শিপিং খরচ ইত্যাদি কমানো যায়। বিশেষ করে ছোট ছোট ইউটিলিটি প্রোগ্রামগুলো যেগুলো দামেও কম সেগুলো এ পদ্ধতিতে বিতরণ করাই সবচেয়ে কার্যকর। রেজিস্ট্রেশন হলে স্বাভাবিকভাবেই ব্যবহারকারী এমন কিছু পাবে যা দিয়ে সে তার প্রোগ্রামটি রেজিস্টার্ড ভার্সন করতে পারবে। সাধারণত এই এমন কিছুটা হলো একটি রেজিস্ট্রেশন নাম্বার। তবে এটি যে কোনো নম্বরই হতে হবে তা নয়— এটি প্রোগ্রাম ডিস্ট্রিবিউটর কর্তৃক ব্যবহারকারী প্রদত্ত একটি তথ্যও। আপনার প্রোগ্রামের এই রেজিস্ট্রেশন নাম্বারটি অনেক বড় বা আনরিডেবল হওয়া উচিত নয়। একটি

উদাহরণ দেখুন— 048CD791BECX 29240B7C9YX\$@45811F5C6 এটি একটি সত্যিকার প্রোগ্রাম থেকে নেয়া উদাহরণ। কিন্তু আপনার রেজিস্ট্রেশন কী-টি করুন পঠনযোগ্য, যেমন, শুধু সংখ্যা দিয়ে তৈরি করুন। তবে তা সেপারেটেড যেমন : 214-527-917-988-740. আর যদি ওয়েব বেইজড রেজিস্ট্রেশন হয়ে থাকে তাহলে সমস্যাটির সমাধান করা যায় প্রেসেসটি অটোমেটেড করে অর্থাৎ অনলাইনেই প্রোগ্রামটিকে রেজিস্টার্ড ভার্সনে রূপান্তর করে।

## ক্রয় করার উপায়সমূহ এবং তার প্রতিকার

বেশিরভাগ শেয়ারওয়্যার প্রোগ্রামই খুবই সহজে ক্রয়কৃত হয় এ কারণে যে, এর প্রোগ্রামার হ্যাকার/ক্রয়কারীদের ব্যবহৃত পদ্ধতি সম্পর্কে কোনো ধারণা রাখেন না। এখন আমরা সহজ ও সবচেয়ে বেশি ব্যবহৃত কিছু ক্রয়ক্রয়-এর উপায়ে আলোচনা করব—

## ডেট লিমিটেশন যুক্ত প্রোগ্রাম ক্রয়ক্রয়

লিমিটেড টাইমের ট্রায়াল ভার্সন ব্যবহার করতে দেয় যেসব প্রোগ্রাম সেগুলো অনায়াসেই ভেঙে ফেলে তৎপর ক্রয়কাররা। এমনকি স্মার্ট ব্যবহারকারীদের জন্য বিভিন্ন উপায়ও বাতলে দেয় তারা। যেমন ধরুন এমন একটি প্রোগ্রাম ইন্সটল করার আগে আপনি পিসির তারিখরটি ১২/৩১/২০২০ করে নিলেন তাহলে প্রোগ্রামটি যদি একমাসের ট্রায়াল ভার্সন হয় আর এর প্রটেকশন সিস্টেম সিদ্ধান্ত নেবে ০১/৩০/২০২১ পর্যন্ত প্রোগ্রামটি চলবে। এরপর আপনি আবার সঠিক ডেট সেট করে নিলেন। এ ধরনের চালিকা আজকাল প্রোগ্রামাররা কাটিয়ে উঠেছেন। এ জন্য আপনি প্রোগ্রামার হলে যা করতে পারেন তা হলো— পিসির বর্তমান তারিখ যদি অনেক বড় হয় (যেমন— আপনার প্রোগ্রাম রিলিজ ডেট হতে ছয় মাসের বেশি), এরর মেসেজ দিন নতুন ভার্সন ব্যবহার করার জন্য। এর চেয়েও কার্যকরী হবে আপনার প্রোগ্রামটি যতবার রান করানো হবে ততবারই এই চেকটি করানো। ধরুন প্রোগ্রামটি হঠাৎ একবার পেল যে কারেন্ট ডেট আসলে পুরনো তাহলে এমন ধরনের মেসেজ দিন যাতে ব্যবহারকারী ভাববে আপনার প্রোগ্রামটি এবার আনইন্সটল করা উচিত। কিন্তু ব্যাপারটি যদি ব্যবহারকারীর ইচ্ছাকৃত না হয়? অর্থাৎ নিরীহ গোবেচারার ইউজারের CMOS ব্যাটারি ফুরিয়ে যাওয়ার ধরুন হঠাৎ তার পিসিতে ১৯৯৮ সাল হয়ে গেছে! তাই এটা করা উচিত হবে না। পদার্থবিদ্যার বৃদ্ধাঙ্গুলি নীতির মতে, মনে রাখবেন আপনার প্রোগ্রামের প্রটেকশন যদি এতটা রুঢ় হয় তাহলে আপনার প্রোগ্রামিং দক্ষতা সম্বন্ধে ব্যবহারকারীর

বিরূপ ধারণা জন্মাবে তাই সে আপনার প্রোগ্রামটি ক্রয় করতে ইচ্ছুক হবে না। এমন ক্ষেত্রে আপনি ব্যবহারকারীকে তারিখটি ঠিক করে দিতে বলুন (আপনার প্রোগ্রাম)।

আপনি যদি ইন্টারনেট বেসড প্রোগ্রাম ডেভেলপার হয়ে থাকেন তাহলে আপনি অনলাইন টাইম সার্ভারের সহায়তা নিতে পারেন। যেহেতু আপনার প্রোগ্রাম ব্যবহারকারী অনলাইনেই কাজ করবে তাই আপনার প্রোগ্রামের সময়-তারিখ একটি রিলায়েবল সার্ভারের ডেট টাইমের সাথে মিলিয়ে নিতে পারেন। এর প্রয়োগক্ষেত্র অবশ্যই সীমিত কিন্তু এই পদ্ধতি অনেকেই আজকাল ব্যবহার করছেন।

সঠিক কারেন্ট ডেটটি জানা সম্ভব না হলেও তারিখ যে পরিবর্তন করা হয়েছে তা সনাক্ত করা সম্ভব। অনেক প্রোগ্রামারই তাদের সম্বন্ধীয় তথ্য উইন্ডোজ রেজিস্ট্রিতে রাখে। তাই রেজিস্ট্রি পরিবর্তনের তারিখটি চালু হয়

প্রোগ্রামটি কেবল ব্যবহার করতে পারার জন্য উইন্ডোজ রি-ইন্সটল করতে হবে। এখন আপনার কল্পনাকে কাজে লাগান ক্রয়কার হিসেবে আপনার আনইন্সটলার যাতে এ ধরনের চিহ্ন

কিছুতেই মুছে না ফেলে। সবচেয়ে গুরুত্বপূর্ণ হলো— আপনার প্রোগ্রামটি যে এই চিহ্ন কোথায় রেখেছে তা যেন ক্রয়কারের খুঁজে পাওয়া দুঃসাধ্য হয়। এক্ষেত্রে রেজিস্ট্রিতে তথ্য রাখার বদলে, উইন্ডোজ বা সিস্টেম ফোল্ডারে কোনো dll ফাইলে রেখে দিতে পারেন, যা কিনা আন ইন্সটলার রিমুভ করবে না।

আরেক ধরনের প্রটেকশন আছে যা কিনা প্রোগ্রাম রানিং এর কাউন্ট-এর ওপর নির্ভর করে। যেমন— ইয়াক কলান্স-এর ডাউনলোড ভার্সনটি আপনি মাত্র ১৯ বার খেলতে পারবেন। এটা কীভাবে করা হয়? এটা খুবই সম্ভব যে এটি কোনো রেজিস্ট্রি লোকেশন অথবা .ini ফাইলে কাউন্টারটি রাখা হয়। এবং এই ভ্যালুটি কাউন্টার নেমটি অবশ্যই



তখন C:\windows\system.dao আর C:\windows\user.dao নামে রেজিস্ট্রির দুটি হিডেন ব্যাকআপ তৈরি করে। সুতরাং এই ফাইলদ্বয়ের লাস্ট মডিফিকেশন তারিখ যদি কারেন্ট ডেট-এর চেয়ে বেশি হয়, তাহলে বোঝা যাবে যে, ব্যবহারকারী হয়তো ব্যাক ডেট করার চালাকিটা করেছে। এই সাথে কিন্তু রেজিস্ট্রির ব্যাকআপ রেজিস্ট্রিরও তারিখ মেলাতে হবে কারণ মডিফিকেশন ডেট যদি কারেন্ট ডেট ডেটটিই হয়! উইন্ডোজ কোনো ফাইলের শুধু তৈরি তারিখ স্টোর করে না; এর লাস্ট এক্সেস তারিখ স্টোর করে রাখে। এটি থেকেও কারেন্ট ডেট জানা যেতে পারে যেমন— Autoexec.bat ফাইলটি প্রতিবারই এক্সেস হয়। এখনকার প্রোগ্রামগুলোর সাধারণত ইন্সটলারও ডিইন্সটলার থাকে। আপনার ইন্সটলারটি ব্যবহারকারী পিসিতে (উইন্ডোজ রেজিস্ট্রিতে) এমন কিছু চিহ্ন রেখে দিতে পারে যাতে প্রোগ্রামটি যে ইন্সটল করা হয়েছিল তা বুঝা যাবে এবং ইন্সটলার চলতে চাইবে না। তবে এক্ষেত্রে আপনার প্রোগ্রামটি এমন দক্ষ হতে হবে যে ব্যবহারকারীকে আপনার

এনক্রিপ্টেড আকারে রাখা দরকার। বিশেষ করে এটা জরুরি যে, ওটা যে কাউন্টার তা যাতে কিছুতেই না বোঝা যায়। এ ধরনের প্রোগ্রামও ক্রয়কৃত হচ্ছ হরহামেশাই। যদি আপনার প্রোগ্রামটি ইন্টারনেট বেসড হয়, সেক্ষেত্রে কাউন্টারটি লোকাল কম্পিউটারে রাখার বদলে একটি সার্ভারে রেখে দিতে পারেন। এটি আরো জটিল করে দিতে পারেন যদি কাউন্টারসহ অন্যান্য প্যারামিটারের একটি চেকসামও সংরক্ষণের ব্যবস্থা করেন। সেক্ষেত্রে সম্পূর্ণ প্রোগ্রামের সেটিং প্যারামিটার কেবল একভাবেই রিস্টোর করা যাবে। যেহেতু রেজিস্ট্রি তথ্য রাখার জন্য কমন জায়গা হয়ে যায় সেহেতু আপনি একটি .ini ফাইলে এটি করতে পারেন। (অবশ্য প্রটেকশন .ini-র বদলে অন্যকিছু হতে পারে) এর মধ্যে সেটিংস রাখুন সেই সাথে কাউন্টার ভ্যারিয়েবলটিও। এগুলোর নাম অবশ্যই জটিল করে দিতে হবে। ফাইলটিকে উইন্ডোজ অথবা সিস্টেম প্রোপার্টিতে রাখুন আর অতিঅবশ্যই যা করতে হবে তা হলো— প্রতিবার কাউন্টারটির মান পরিবর্তনের পর ফাইলটির প্রথম তৈরির সময়কার

ডেটাই সেট করে দিতে হবে। এতক্ষণ এন্টি হ্যাকিং বিষয় নিয়ে পড়ে নিশ্চয়ই বুঝতে পারছেন কেন তারিখ পরিবর্তন করা যাবে না? যাবে না এই জন্য যে, তারিখ পরিবর্তন হলে সাধারণ ক্র্যাকাররা অতি সহজেই অনুমান করে নেবে যে, এটিই তাদের গবেষণা করার টার্গেট ফাইল।

## হ্যাকিং বা ক্র্যাকিং-এর কতিপয় পদ্ধতি

অনেক প্রোগ্রামই আছে যাতে রেজিস্ট্রেশন নম্বর ইউজার নেম-এর ওপর নির্ভর করে। অনেকক্ষেত্রে এ ধরনের প্রোগ্রাম বাজারে ছাড়া মাত্রই তার জন্য অনেক কী জেনারেটিং প্রোগ্রাম হাজির হয়ে যায়। এ প্রোগ্রামগুলো যে-কোনো নামের জন্যই কী জেনারেট করতে পারে। তবে এ ধরনের কী জেনারেশন প্রোগ্রামও অসম্ভব করে দেয়া যায় উন্নত এনক্রিপশন ব্যবহার করে। এর জন্য আপনি হয়তো একটি সাধারণ এ্যাসিমেন্ট্রিক এনক্রিপশন সম্পর্কে বিশদ জানতে [www.rsa.com](http://www.rsa.com)-এ টু মেরে আসতে পারেন। এক কথায় বলতে গেলে বলা যায়, এই এনক্রিপশনের

আজকালকার বেশিরভাগ প্রোগ্রামই (নন-শেয়ারওয়্যারগুলো) ইন্সটলেশন-এর সময় একটি সিরিয়াল নম্বর চায়। বৈধ ব্যবহারকারীর জন্য এই সিরিয়াল নম্বর থাকে তার কাছে বিক্রি করা প্রোগ্রাম ও এর ডকুমেন্টসমূহের সাথে। কিন্তু পাইরেটেড প্রোগ্রামের ক্ষেত্রে কী হয়—ইন্সটলার সিডিতে পাইরেটের সিরিয়ালটি দিয়ে দেয়। এই সিরিয়াল নম্বর এমনভাবে তৈরি হওয়া উচিত যাতে পাইরেটদের অনুমিত কোনো ফর্মুলা এতে কাজ না লাগে যেমন—কোনো র্যানডম নম্বর যাতে কাজ না করে। মাইক্রোসফট এর আগে সফটওয়্যারগুলো কিছুদিন আগেও সিরিয়াল নম্বর হিসেবে '1' দিয়ে ভরিয়ে দিলেই কাজ করত। অবশ্য আমার ধারণা মাইক্রোসফট এই ফাঁকটি ইচ্ছা করেই রেখেছিল যাতে বিনামূল্যে পেয়ে তাদের প্রোগ্রামের ব্যবহার বেড়ে যায়। আদতে তো তাই হয়েছিল। তবে সিরিয়াল নম্বরের বদলে আরেকটি কার্যকর প্রটেকশন ছিল এরকম : অমুক নং পেজের অমুক নং শব্দগুলো যেখানে কী-ওয়ার্ড হিসেবে কাজ করবে। এবারে কিছু কথা আপগ্রেড প্রোগ্রাম

একটি আইডিয়া হতে পারে যে এটি প্রয়োজনীয় এমন কিছু রেজিস্ট্রি সেটিং করবে না যেটি এর পূর্বতন ভার্সনেরই সেট করার কথা। এর চেয়েও জটিল জটিল উপায় হয়তো আপনিও ভেবে বের করতে পারেন। তবে জেনে অবাক হবেন হ্যাঁকার বা ক্র্যাকার এতই প্রতিভাবান যে, তারা এমন কোনো পথ নেই যেটি ধরে চিন্তা করে না। সে কারণেই আমরা বাজারে পাইরেটেড ভার্সনে সব ধরনের প্রোগ্রামই পাচ্ছি।

## কোড মডিফিকেশনের মাধ্যমে হ্যাকিং

এখন আমরা সাধারণভাবে কিছু হ্যাকিং কৌশল দেখব যেগুলো ব্যবহার করে তৈয়ারি হ্যাকিং করেন। তবে আবেগে আপুত হবেন না কেননা এখানে পুরো বিবরণও নেই আর যেখানে আমার উদ্দেশ্য হচ্ছে পাইরেসি প্রতিরোধ কৌশল। জেনে অবাক হবেন বেশিরভাগ শেয়ারওয়্যার প্রোগ্রামকেই ফুল-স্কেল ভার্সনে রূপান্তর ঘটাতো অনেক সময় মাত্র একটি বাইট পরিবর্তন করে কয়েক মিনিটেই সেটিকে পাইরেট করে ফেলে ক্র্যাকারগণ। তাহলে বুঝতেই পারছেন কেন তারা প্রোগ্রাম লেখে না। লেখে তো অবশ্যই তবে তা ক্র্যাকিং-এর খাতিরেই। এখন আপনি নিশ্চয়ই ভাবছেন যে কীভাবে হ্যাঁকাররা একটি মাল্টি মেগাবাইট প্রোগ্রামকে মাত্র কয়েক মিনিটেই ফর্ডাফাই করে ফেলে? (এটি কিন্তু কাল্পনিক নয়, সত্যিসত্যিই তারা তা করে থাকে)। সত্যি বলতে কি এই দ্রুত হ্যাকিং-এর ক্ষেত্রে নিরীহ প্রোগ্রামাররাই তাদের হেল্প করে থাকে। চলুন একটি উদাহরণ দেখা যাক একটি কাল্পনিক প্রোগ্রামের—

```

BOOL isRegistered ()
{
...
return TRUE or FALSE;
/* প্রোগ্রাম রেজিস্টার্ড কি না তা পরীক্ষা করার জন্য লাগসই একটি কোড এখানে থাকতে পারে/

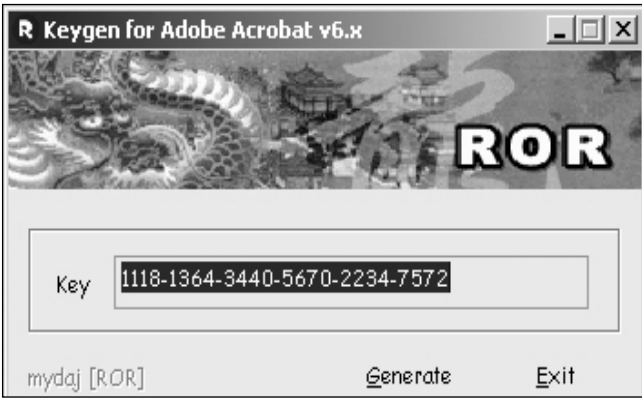
}
//Warning users to register.
if(!isRegistered()) MessageBox
("Not Registered!", "Register Plesed");
//verify if registered to allow some functions.
function if(isRegistered()) allow this Operation();
else
MesaageBox("Unregistered

```

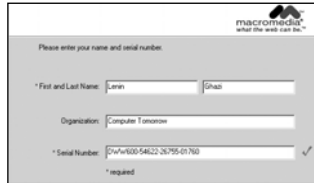
copy!"; This function is only for registered version only"); বেসিক প্রোটেকশনের জন্য এটি ভালোই কার্যকর। তাছাড়া isRegistered() ফাংশনটি মনে করুন অত্যন্ত শক্তিশালী RSA-এর ওপেন-কী এলগরিদম ব্যবহার করেই কোড করেছেন। এবং ফাংশানের চেংকি ছড়িয়ে আছে বিভিন্ন পর্যায়ে। তাহলে হ্যাঁকার বা পাইরেটেরা-এর কীই বা করতে পারবে? সে হয়তো কেবল Register Please! স্ক্রিংটি খুঁজে বের করবে। কেননা জানেন তো কম্পাইলড কোডে তো আর ইন্সট্রাকশন পড়া সম্ভব নয়। তাহলে চলুন দেখি সে এটি খুঁজে কী পাবে? call xxxxx; function is Registered or eax, eax jnz yyyyy; \* ... push <found\_address>; হয়তো বা এমনি একটি জায়গা হ্যাঁকার ... call zzzzz; Message Box খুঁজে পাবে।

এখন একজন হ্যাঁকার যে একমাত্র বাইটটি পরিবর্তন করবে সেটি হলো স্টার চিহ্ন দেয়া লাইনটির। এটি খুঁজে পেতে তার কয়েক সেকেন্ড মাত্র লাগবে এবং প্রোগ্রামটি কোনো রেজিস্ট্রেশনেরই প্রয়োজন থাকবে না তার মতই (!) কর্মের পর। যদিও এ প্রোগ্রাম বারবার চেক করা হচ্ছে এবং খুবই জটিল এলগরিদমও ব্যবহার করা হচ্ছে কিন্তু হ্যাঁকারকে তা কোনোমতেই ষোল খাওয়াতে পারবে না। এমনকি হ্যাঁকার সামান্য পরিবর্তন করে এর isRegistered() ফাংশনটি যাতে সর্বদাই True রিটার্ন করে তারও ব্যবস্থা করতে পারে।

এই উদাহরণে আমরা দেখতে পাচ্ছি যে, প্রথমত, যদি হ্যাঁকারটি প্রথমেই প্রোগ্রামের ঐ অংশে পৌঁছাতে না পারে তাহলে সে ভালোই গলদঘর্ম হবে। দ্বিতীয়ত, এটা সব সময়ই সত্য নয় যে, হ্যাঁকার মাত্র কয়েকটি বাইট বদলেই একটি প্রোগ্রাম তার আয়ত্তে নিয়ে ফেলবে। আমাদের উদাহরণে মোক্ষম জায়গাটি পাওয়া হ্যাঁকারের জন্য সহজ হয়েছিল কেননা সে একটি মোক্ষম স্ট্রিং পেয়ে গিয়েছিল। তবে ঐ উদাহরণের মতো স্টেম্পট না থেকে কোনো রিসোর্স Id থাকতে পারত। যদি মেসেজকে একটি স্টেম্পট রিসোর্স হিসেবে বা ডায়ালগ বক্স হিসেবে রাখা থাকে। ধরুন এতে কোনো বিশেষ নম্বর থাকল। ধরা যাক ২০। তাহলে হয়তো হ্যাঁকার এরকম একটি কোডের খোঁজ করবে। CMP<something>.20 এরকম সার্চ যাতে সহজে না করা যায় সে জন্য এ ধরনের ব্যাপারগুলোকে ছদ্মবেশে রাখতে হবে। যেমন— স্টেম্পট স্ট্রিংগুলোকে (মেসেজ, ফাইলনেম, রেজিস্ট্রেশন কোড প্রভৃতি) এনক্রিপ্টেড আকারে রাখতে পারেন। আর প্রোগ্রামে



ক্ষেত্রে একটি কী ব্যবহৃত হয় এনক্রিপশনে এবং ডিক্রিপ্ট করতে অন্য একটি কী দরকার হয়। এমনকি একটি কী জানা থাকলেও অন্যটি বের করা সম্ভব নয়। ধরা যাক একটি ইউজার নেম ও রেজিস্ট্রেশন নম্বর নিয়ে এই এ্যাসিমেন্ট্রিক এলগরিদম দিয়ে এনক্রিপ্ট করাতে হবে আর এর এ্যানক্রিপশন পাসওয়ার্ড বা কী-টি কেবল আপনিই জানবেন। আপনার প্রোগ্রামটি রেজিস্ট্রেশন নম্বরকে ডিক্রিপ্ট করে ইউজার নেম-এর রেজাল্ট-এর সাথে মিলিয়ে দেখবে। ডিক্রিপশন কী-টি প্রোগ্রামেই বিল্ট করা থাকবে। বলা হয়ে থাকে এ ধরনের এ্যালগরিদম সঠিকভাবে ব্যবহার করা গেলে কোনো কী জেনারেটিং প্রোগ্রামই তৈরি করা অসম্ভব। কিন্তু আপনি যদি একজন সংক্রেতা হন আর আপনার কেনা প্রোগ্রামের সিরিয়াল নম্বর ও তথ্যাদি যদি কেউ চুরি করে ছড়িয়ে দেয়। সেক্ষেত্রে এই এনক্রিপশন কোনো কাজেই আসবে না। অথবা ক্র্যাকাররা যদি রিভার্স ইঞ্জিনিয়ারিং করে আপনার কোডে রদবদল করে দেয়।



সম্পর্কে। এক্ষেত্রে আপগ্রেড ইন্সটলার যদি নিশ্চিত হয় যে পূর্ববর্তী ভার্সনের একটি ইন্সটলেশন এই পিসিতে ছিল তবেই সেটি ইন্সটলেশন শুরু করবে। যেমন— উইন্ডোজ ৯৫-এর ইন্সটলারটি যদি উইন্ডোজ ৩.১ পেত তবেই সেটি ইন্সটল হতো। কিন্তু জানেন কি, সেই আপগ্রেড ইন্সটলারটি কেবলই windows/system/win386.exe ফাইলটি আছে কি না এমনকি ০ বাইটসও রাখেন তাহলেও সেটি কাজ করবে। তার মানে যে পিসিতে কোনোদিনও পূর্ববর্তী কোনো উইন্ডোজই ছিল না সেটিতেও হ্যাঁকাররা উইন্ডোজ ৯৫ ইন্সটল করার পন্থা বের করল। সুতরাং আপগ্রেড ইন্সটলারকে একটি কৌশলিভাবে তৈরি করা দরকার। এমন

```

কনস্ট্যান্ট থাকলে নিচের মতো করে
করতে পারেন।
define SLY 21062004
int vSly = SLY;
...
Load String
(IDS_SHAREWAREWARNING-
SLY+vSly);
আর, প্রোটেকশনের মডিউলগুলো যাতে
প্রোগ্রামের শুরু দিকেই না থাকে। বেশ
অনেকগুলো টেম্পিং মেথড রাখুন আর
সেগুলো যাতে বারবার চেক করে। আর
চেকআপের পদ্ধতিগুলোও যাতে
পিরিয়ডিক্যালি হয় সেদিকেও খেয়াল
রাখুন।
এবারে আমরা প্রোগ্রামের আরো কিছু
টিপস দেখব যাতে আমাদের প্রোগ্রাম
হ্যাকারদের মাথার ঘাম পায়ে ফেলার
ব্যবস্থা করা যাবে। অন্তত তারা মাত্র
কয়েকটি বাইট এদিক সেদিক করেই
পার পাবে না।
Struct VeryImportant{
এরকম একটি স্ট্রাকচার রাখুন যাতে
কেবল রেজিস্টার্ড ইউজারদের জন্য
ফাংশনগুলোতে দরকার হবে এমন
কনস্ট্যান্টগুলো থাকবে। সেক্ষেত্রে আপনি
সেই ফাংশনে হয়তো a = b + z এর
বদলে a = b + myconst.two;
লিখবেন।
এভাবে করে আপনি যথেষ্ট জটিলতায়
ফেলে দেবেন একজন হ্যাকারকে।
কোনো ফাংশনের পরমেন্টারকেও
কনস্ট্যান্ট হিসেবে ব্যবহার করতে
পারেন, সেক্ষেত্রে এই কনস্ট্যান্ট দিয়েই ঐ
পয়েন্টেড ফাংশনটিকে কল করতে
পারেন।
}myconst={xxxx};
আমাদের এই xxxx কনস্ট্যান্টগুলোকে
ভালো এ্যালগরিদমে এনক্রিপ্ট করে রাখা
উচিত। এমন কোনো সহজ এ্যালগরিদম
ব্যবহার করা উচিত নয় যেগুলো বহুল
পরিচিত। বরং rc2 বা rc4-এর মতো
প্রফেশনাল এ্যালগরিদম ব্যবহার করা
উচিত।
Const MyconstCRC=xxx;
...
int isRegistered()
{
return
CalculateCRC(&myconst)==MyC
onstCRC;
}
...
BYTE regnum [REGNUMLEN];
...
//function to be called once for
init.
Void initProtodion(){
//now decrypte the constants with
regnum or its part.
decrypt (&myconst, regNum);
}
অন্য কোডগুলো আমাদের আগের
কোডের মতোই করতে পারি। এখনো
কিন্তু হ্যাকার মহাশয় এই

```

```

isRegistered() খুঁজে পেতে পারে।
কিন্তু সে এটিকে মডিফাই করলে
প্রোগ্রামটি আর কাজই করবে না।
কেননা কনস্ট্যান্টগুলোতে ঐ রেজিস্ট্রেশন
নাশার ছাড়া ডিক্রিপ্ট হবে না। মোদাকথা
যে সহজ কথাটি আমাদের মনে রাখতে
হবে তা হলো— কোনো কনস্ট্যান্ট-এর
সাথে ম্যাচিং করার বদলে সেটিকে
আমরা ব্যবহারই করব। যেমন আমরা
রেজিস্ট্রেশন কী খুঁজব না তাকে ব্যবহার
করে প্রোগ্রাম চালাব। তাহলে হ্যাকার
সহজে খুঁজে পাবে না কী সেই ভ্যালু।
এবার আসুন আগের এই চিন্তাটা মাথায়
রেখে অন্যরকম একটি উদাহরণের কথা
চিন্তা করি। ধরা যাক, আপনি একটি
ডেমা প্রোগ্রাম লিখেছেন, যেটিতে
৪০টির বেশি রেকর্ড ডাটাবেজ রাখা
পারমিট করবে না। যদি Cuser নামক
স্ট্রাকচার দিয়ে প্রতিটি ইউজারকে
এন্ট্রিস করা হয় তাহলে ডাটাবেজটি
একটি Cuser-এর অ্যারে আছে।
array=nwe array [usernum];
...
if(usernum>4)...;
এরকম হবার কথা কোডটি। আর
হ্যাকার শুধু খুঁজবে
cmp <???-??>, 40
তাহলে একে একটু জটিল করে দেবার
জন্য আমরা ঐ লাইনটিতে লিখতে
পারি—
if (usernum*2+1>81)...;
জটিলতা আরো বাড়তে আপনি নিচের
মতোও করতে পারেন—
Cuser *array;
//in a registered version initialize
with a different function :
void initUserArray()
{
int x= —83, y=3;
array = new array [(x+y)-2];
}
...
if(usernum*2+1>81)...;/**
এক্ষেত্রে এই ডবল স্টার দেয়া লাইনটি
যদি হ্যাকার খুঁজেও পায়, সে কিন্তু
অন্যায়সে এগোতে পারবে না। অর্থাৎ
তার এটা খুঁজে পেতে কষ্ট হবে কোথায়
আছে সেই ৪০। তবে সমস্যাকে পাজল
বা রিডল হিসেবে নিয়ে তা সমাধান করা
হ্যাকারদের একটি শখের ব্যাপার সূতরাং
এত তাড়াতাড়ি স্বস্তির নিঃশ্বাস ফেলবেন
না। পদে পদে রয়েছে বিপদ।
আনরেজিস্টার্ড প্রোগ্রাম, যেগুলো বারবার
রিমাইন্ডার দেয় সেগুলোকে হ্যাকারদের
হাত থেকে নিরাপদ করা খুবই কঠিন।
এটির ক্ষেত্রে হ্যাকার সাধারণত কিছু
এক্সট্রা কোড ঐ প্রোগ্রামটি থেকে খেঁড়ে
ফেলে দেয়। আর আমরা তো জানিই,
গড়ার চেয়ে ভাঙা অনেক সহজ।
রিমাইন্ডার শো করবে এমন প্রোগ্রামের
ক্ষেত্রে, যেখানে রিমাইন্ডার দেখায়
কোনো ডায়ালগ বক্স— যখন
WM_INIT মেসেজ হ্যান্ডলার (ডায়ালগ

```

ইনিশিয়ালজ করে যেটি) রান করে তখন একটি টাইমার এক সেকেন্ড হিসেবে সেট করতে পারেন। আর যখন WM\_TIME নামক টাইমার হ্যান্ডলারটি রান করবে তখন আপনি কিছু প্রয়োজনীয় ভারিয়েবলটি সেট করতে পারেন অথবা এমন কিছু অপারেশন করতে পারেন যা পরবর্তী ধাপে দরকারি। যদি হ্যাকার কিছু না দেখেই ডায়ালগটি শো করা বন্ধ করে তাহলে প্রোগ্রামটিও চলতে চাইবে না। আরেকটি উপায় হচ্ছে আপনার প্রোগ্রামের ইন্ট্রিটি চেক করা। অর্থাৎ হ্যাকার কোনো মডিফাই করল কি-না তা খুঁজে বের করা। এটা আপনার প্রোগ্রামের গতি একটু স্লথ করে দিতে পারে তবে ভাইরাস ইনফেকশন বা ইন্ট্রিটি চেক-এ কোনোরকম ক্রটি বের করতে কাজে দেবে। যদিও হ্যাকার মহাশয় এই ইন্ট্রিটি চেক করার মড্যুলটিই লাপাতা করে দিতে পারে প্রোগ্রাম থেকে। বুঝতেই পারছেন এই চোর পুলিশ খেলার শেষ নেই বললেই চলে।

### কপি প্রোটেকশন

এখন আলোচনা করব আরেকটি গুরুত্বপূর্ণ ফিচার নিয়ে। সেটি হচ্ছে কপি প্রটেকশন। ওপরের ব্যাপারগুলো মেনেও আপনার প্রোগ্রামটি এমন হতে পারে যে একজন বৈধ ব্যবহারকারীর পিসি থেকে কেউ ইন্সটলড কপি চুরি করে কপি করে নিল। আমরা কিছুবহুল ব্যবহৃত টেকনিক সম্পর্কে জানব।

### কী ডিস্ক

এই পস্থা এখন বড় বেশি সেকেন্দে। প্রথমত, বৈধ ব্যবহারকারীর জন্য এটি একটি বোমা বৈকি। দ্বিতীয়ত, ডিস্ক-এর ভরসা কয় সেকেন্ডের ? আবার অনেকে মনে করেন যে যে-কোনো ডিস্কই কপি করা যায় বিশেষ প্রোগ্রাম দ্বারা তাই পদ্ধতিটি অস্বাভাবিক ছিল না ব্যাপারটি কিন্তু সত্য নয়।

### ইলেকট্রনিক কী

এটার যেমন আছে সুবিধা তেমনি অসুবিধা। যেমন কেউ এই ধরনের নিরাপত্তাওয়াল দুটি বা তার বেশি প্রোগ্রাম ব্যবহার করেন। সেক্ষেত্রে এই কী বা দঙ্গল ব্যবহার বন্ধি বটে আর আপনার প্রোগ্রামটির দাম যদি কী-টির চেয়ে কম হয় তাহলে কী করবেন ? আর হ্যাকাররা এমন ইমুলেটেড কী তৈরি করতেও সিদ্ধহস্ত অর্থাৎ প্রোগ্রামের ঐ রুটিনকে আসলে কোনো কী পড়তেই দেবে না বরং রিডিবেক্ট করা একটি আউটপুট পড়িয়ে দেবে।

### হার্ডওয়্যার বাইন্ডিং

হার্ডওয়্যার বাইন্ডিং হলো, কোনো সফটওয়্যারকে একটি নির্দিষ্ট কনফিগারেশনের মেশিনে ইনস্টল করতে দেয়া। কিন্তু এটিও সম্ভব নয় কেন না

ইউজার প্রতিনিয়তই হার্ডওয়্যার পরিবর্তন, আপড্রেড ইত্যাদি করে থাকে। তবে একসময় যখন লোকে অনেক খরচ বলে বছরের পর বছর আপড্রেড করতে না তখন এটি বেশ কার্যকর ছিল।

### সফটওয়্যার বাইন্ডিং

এটিও খুব ফলদায়ক নয় বরং ব্যবহারকারীর ঝড়ই বাড়ায় এটি। তাছাড়া ইন্সটলারের আকারও এতে বেড়ে যেতে পারে।

### উপসংহার

আপনার প্রোগ্রামটি যদি সিডিতে বিতরণের জন্য হয়। আর যদি খুব বড় না হয় তাহলে সিডিটিকে ইন্সটলেশনের জন্য দরকার এমন তথ্য দিয়ে ভরিয়ে ফেলতে পারেন। যদি কাজটি দক্ষতার সাথে করতে পারেন তাহলে আপনার ছোট প্রোগ্রামের জন্য একটি সিডি পুরো খরচ করতে পাইরেটদের বাধবে। কেননা পাইরেটদের কাজই হলো একটি সিডিতে যত বেশি পারা যায় সফটওয়্যার লোড করে দেয়া। সূতরাং আপনি চাইলে বোনাস হিসেবে অন্যান্য প্রোগ্রামও রাখতে পারেন সিডিটিতে। পাইরেটরা যাতে সেগুলো বাদ দিতে না পারে তাই গুলোর উপস্থিতি চেক করার ব্যবস্থা রাখুন। আবার এও চেক করতে পারেন যে নতুন কোনো প্রোগ্রাম যা আপনি দেন নি তা সিডিতে রাখা হয়েছে কি না। শেষ করার আগে আরেকটি গুরুত্বপূর্ণ ব্যাপারে দৃষ্টি আকর্ষণ করছি। এই ফিচারের এন্ট্রি হ্যাকিং তাড়নায় উজ্জ্বলিত হয়ে আপনি হয়তো খুবই সতর্কতার সাথে জটিল ও দুর্লভ্য টেকনিক ব্যবহার করলেন। ধরা যাক— ঐ যে কাউন্টার-এর কথা বলেছি সেই কাউন্টার হয়তো .ini ফাইলে রাখার বদলে আপনি ভাবলেন উইন্ডোজ API ব্যবহার করে WritePrivateProfileString ব্যবহার করলেন। বিশেষ করে আমরা যে ফাইলের ডেট অপরিবর্তিত রাখার কথা আলোচনা করেছি সেটি কিন্তু গুরুত্বপূর্ণ ছিল। আপনারা হয়তো জানেন যে, উইন্ডোজ .ini ফাইলগুলোকে স্পিড বাড়ানোর খাতিরে যদি উইন্ডোজ ঐ .ini ফাইলটি সেভ করতে দেরি করে ফেলে এরপর মডিফিকেশনের একটি অবাঞ্ছনীয় তারিখ সেট করে ফেলে তাহলে কিন্তু ঘোর বিপদ। এখানে বোঝা যাচ্ছে কতটা রিটে হতে হবে প্রোগ্রামিং।

### পরিশেষ

আপনার সব পরিশ্রমই বৃথা যাবে যদি আপনার প্রোগ্রাম কেউ কিনল না অথচ সবার কাছেই এর পাইরেটেড বা ক্র্যাকড কপি পৌছে যায়।

■ গাজী লেনিন  
ghazilenin@hotmail.com