Doctor, doctor, this guy, he's hurt really bad.
WHAT DO WE KNOW ABOUT HIM?
That's the problem, we don't know a thing. No name, no previous health records, heck, no English.
WHAT ARE WE GONNA DO!?!?
WAIT!
He's Got
THE
SAFEBYTE
With all this info, we're hours ahead of schedule. Let's get started!
[MUMBLING]
THANKS SAFEBYTE...
IPRO 304
SafeByte Health System
Portable and Secure Data Storage System

Physical X-Rays/EKG’s
Paper Health Records
Doctor/Pharmacist Input

SafeByte
Names.... blah
What are we doing?

The goal of this project is to design software to support the secure use of portable memory as a medium for transferring secure data.
Objectives

Our Product will be:

- Portable
- Secure/Accurate
- Method of Authentication for software and data
- Cost-effective
- Easy to use
- Capable of having multiple users
- Compatible
Electronic Health Records
Benefits of EHR

- Immediate and Universal access
- Easier and quicker navigation
- No lost charts
- Standardization among health care providers
- Reduction of paperwork and documentation error
- Ability to transmit information to other providers
Why the delay?

- Privacy of patients record
- Lack of uniform Standards
- Time constraints
- For patients or insurers?
- Funding
HIPAA

Health Insurance Portability and Accountability Act
Where Did HIPAA come from?
Who sees the Medical Records?
Guidelines to Being HIPAA Compliant.
How it relates to our Project?

- Privacy of electronic records
  - Levels of Security~ Accessing and Editing
    - Doctors, Patients, Pharmacists, Insurance...etc.
  - Emergency Information
  - Access Logs~ Time stamps and Signatures
- HIPAA Compliancy
About the Program

- Eliminates paper records
- Various types of Medical Information
- Multiple Users
- User Access Levels
## User Matrix

<table>
<thead>
<tr>
<th></th>
<th>Patients</th>
<th>Doctors</th>
<th>Pharmacists</th>
<th>Emergency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>See</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Edit</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Personal Information</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Allergies and Illnesses</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Medications</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Vaccinations</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
**Personal:**
Lutfi Dughman
yadda yadda yadda

Back
Physical Interface Security
Physical Interface Security

Problem to Address:

- The flash drive must not have the ability to attack the host computer:
  - Disable the Autorun Feature in Windows
  - Identify the Correct Device for a Virus Scan
Physical Interface Security

Research:

- Identified Possible Flash Drive Threats
- Determined Solution Possibilities
- Chose the Most Feasible and Effective Solutions
Physical Interface Security

Solution:

Disabling Autorun
- Editing the Registry

Identifying Devices
- Device Scan
- Identification File
- Third Party Virus Scan
Physical Interface Security

2 of 3 Initial Layers of Defense:

- Autorun Prevention
  - Guarantees safety from a virus executing itself

- Virus Scan
  - Can ensure user that a virus is not embedded on the drive
Application Security:
Application Authentication
Team
Why Authenticate SafeByte Software?

- Authentic software is important for any application that stores & manipulates a user’s private information.

Unauthentic SafeByte Software can result from:
- Viruses
- Hackers
- File errors

GOAL:
A practical way of checking if the SafeByte software is authentic....
Solution: Digital Signatures!

What is a Digital signature?

A Digital Signature is an electronic (digital) stamp or seal that is appended to data during data exchange.

How does it work?

- The slightest change to the SafeByte Medical Application will produce a very different signature.

- Signatures are easy to use for comparison, as opposed to using the entire application.
How SafeByte uses Digital Signatures:

- Authentic SafeByte software is passed through a signature generator.
- The result is Authentic SafeByte Digital Signature.
- Later on, another signature is generated by the user.
- The current SafeByte software is verified by comparing the current and authentic signatures.
Implementation

In Flash Drive…

**SafeByte Application**

```
......10110101101010110101010
```

**Signature Generator: MD5**

**Optional:**

On World Wide Web...

**MD5 algorithm.**
MD5 can produce a unique 128 bit “digest” of an input of arbitrary length.

```
= Authentic Signature
```

```
= Current Signature
```

```
= Authentic Signature
```

- True, then application is authentic.
- False, then application is not authentic, & not secure
THE FUTURE?!?!?!?

- Will be able to verify the identities of Doctors, etc. using a third party source (internet)

- Will be able to store wider range of Medical Records
That’s it!