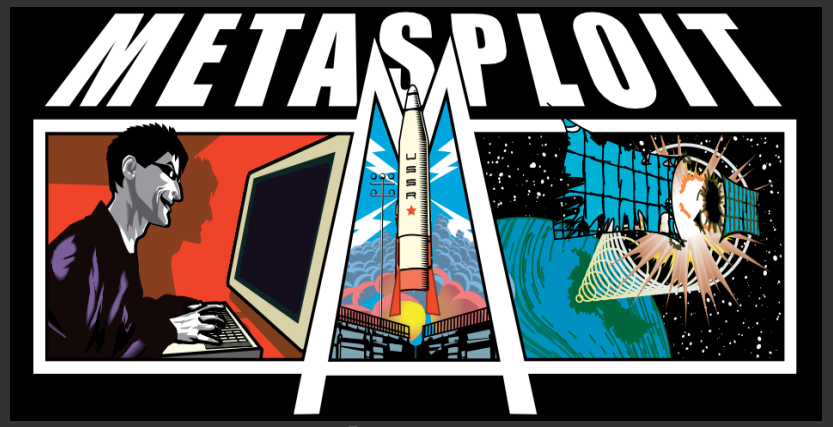
# Attacking .NET Applications at Runtime

Jon McCoy - 2010
www.DigitalBodyGuard.com

# What will this presentation cover?

- How to pWN closed-source .NET applications in new and dynamic ways
- New tools I am releasing
- Show how incredibly vulnerable .NET applications are

# What tools will you get?




- New Metasploit payload
- Tools to do reconnaissance, on the structure of .NET programs
- Beta - Decompilation Tool targeted at .NET Applications protected by wrappers/shells

# What will the hack do?

- Gain access to a target application
  - Access the Object structure
    - Compromise the GUI
    - Subvert core logic
    - Instantiate new features

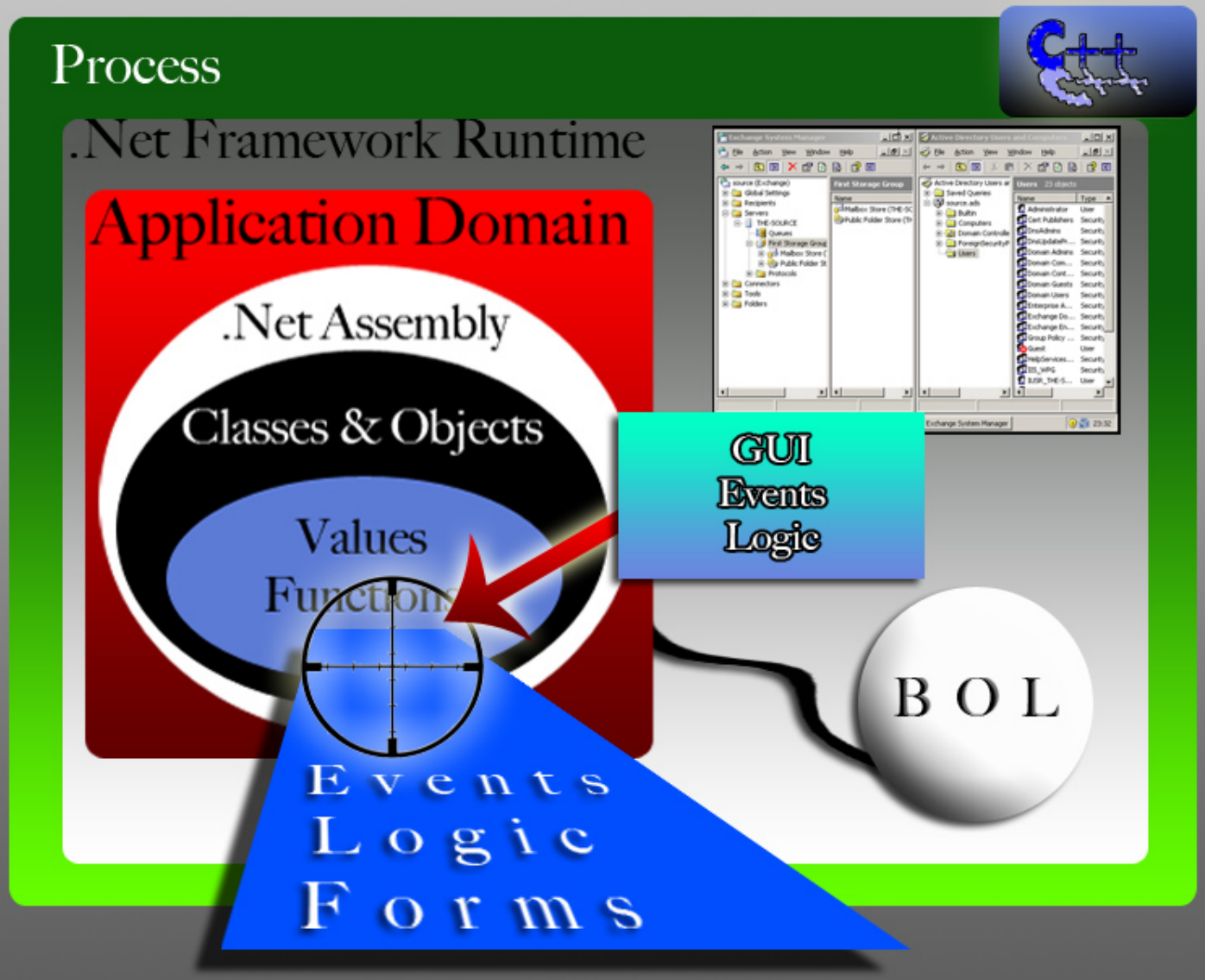
# Connect to the Target

- Inject - Put your code into the target
- Infect - Change the target's code
- Exploit - Take advantage of a flaw
- Attack The Framework - Compromise the framework

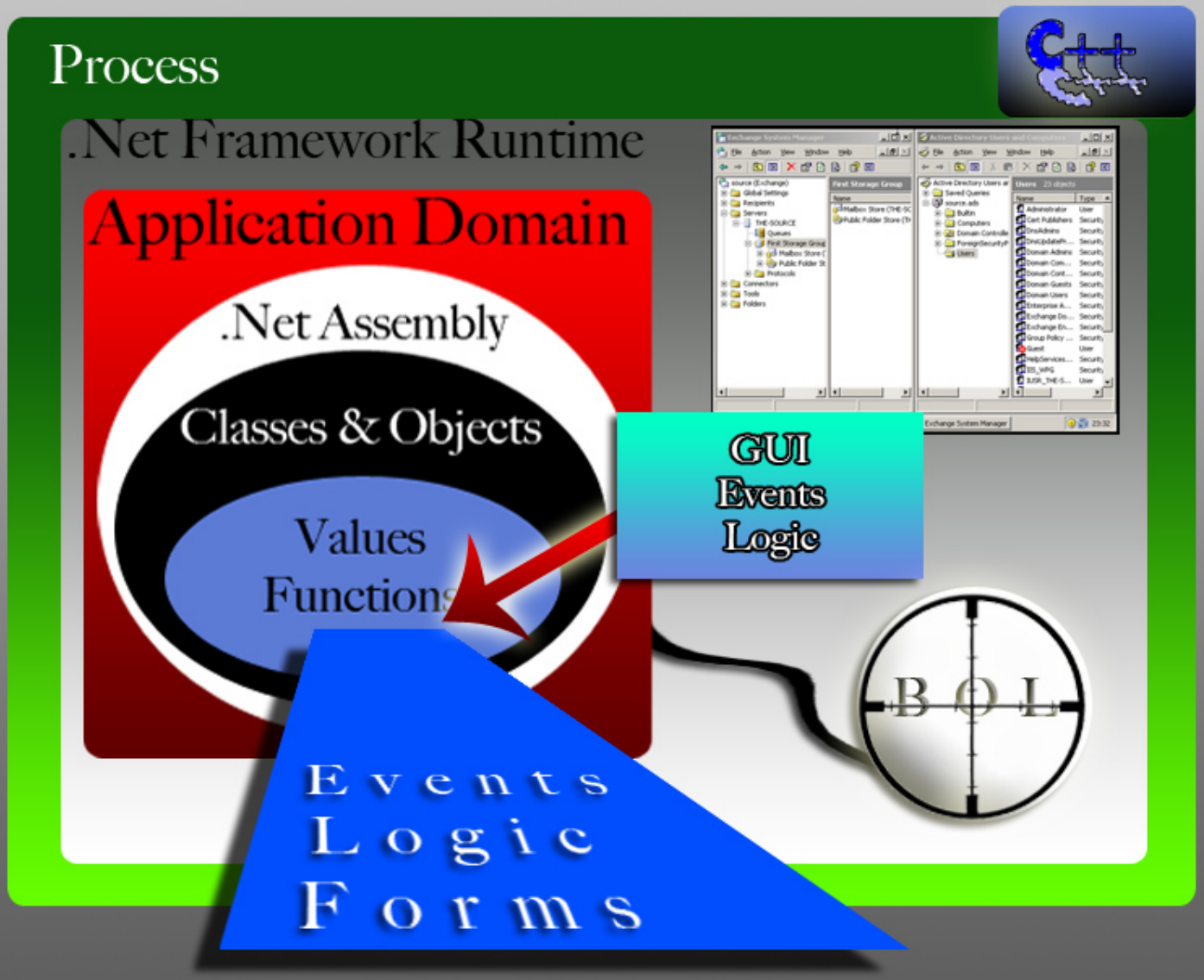
# What we are attacking

- .NET Process
  - AppDomains
    - Assemblies
      - Modules
        - Classes
          - Functionality
          - Objects
            - Events
            - Values
            - Instances

# How .NET Apps Execute at RunTime



# How .NET Apps Execute at RunTime



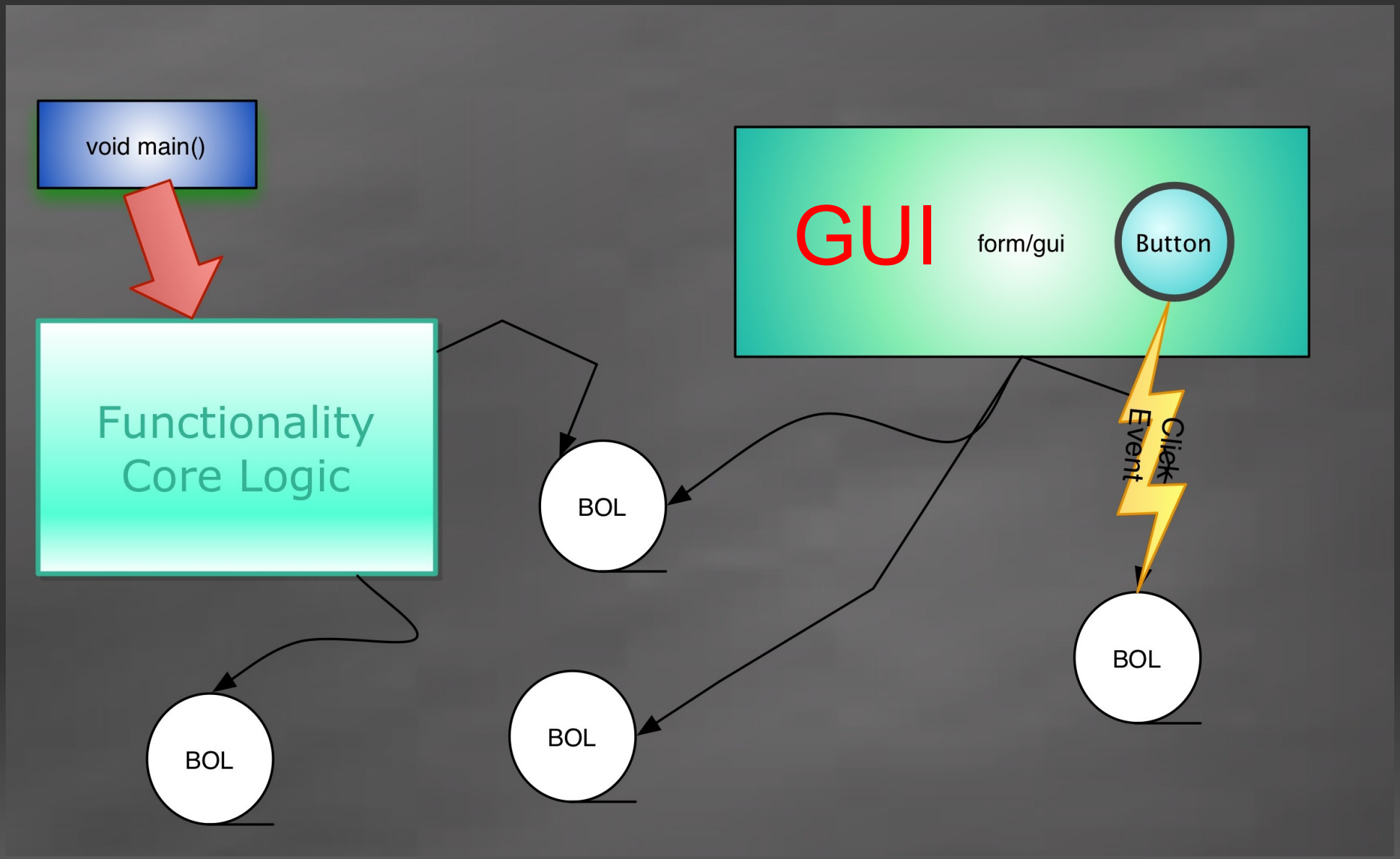
# How the hack works: Overview

1. Connect to the target application
   -Connect With Injection

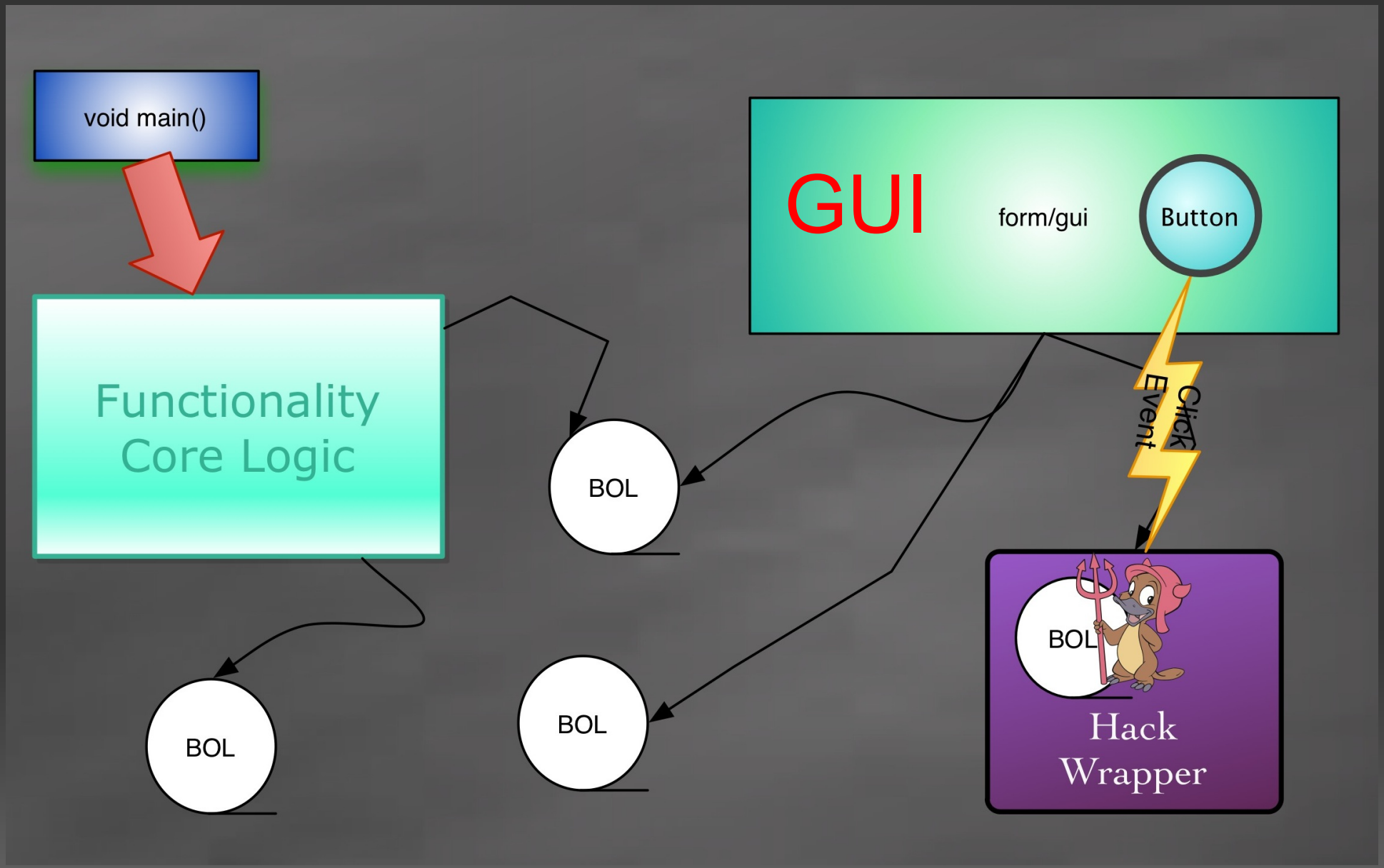
2. Access targets Object structure
   -Move around with Reflection

3. Modify values and/or Objects
   -Modify Objects with Reflection

# Normal Runtime Object Structure



# Hacked Object Runtime Structure



# Sample Code: Hack Event

```
public static void clearClickEvent(System.ComponentModel.Component targetIN)
{
    // flag for reflection on the object targeted by reflection
    System.Reflection.BindingFlags flagOfObject
    = System.Reflection.BindingFlags.Instance | System.Reflection.BindingFlags.NonPublic;

    // get the "events" field on the target
    System.Reflection.FieldInfo FieldEvent;
    FieldEvent = typeof(System.ComponentModel.Component).GetField("events", flagOfObject);
    System.ComponentModel.EventHandlerList R_eventList;
    R_eventList = FieldEvent.GetValue(targetIN) as System.ComponentModel.EventHandlerList;

    // get the "head" field, type is {System.ComponentModel.EventHandlerList.ListEntry}
    // this is not a public type so it can only be refrinced at run time
    System.Reflection.FieldInfo FieldHead;
    FieldHead = typeof(System.ComponentModel.EventHandlerList).GetField("head", flagOfObject);
    object R_head = FieldHead.GetValue(R_eventList);

    // get the "handler" field on the target
    System.Reflection.FieldInfo FieldHandler;
    FieldHandler = R_head.GetType().GetField("handler", flagOfObject);

    // set value of the event head pointer, clear the event list
    FieldHandler.SetValue(R_head, null);
}
```

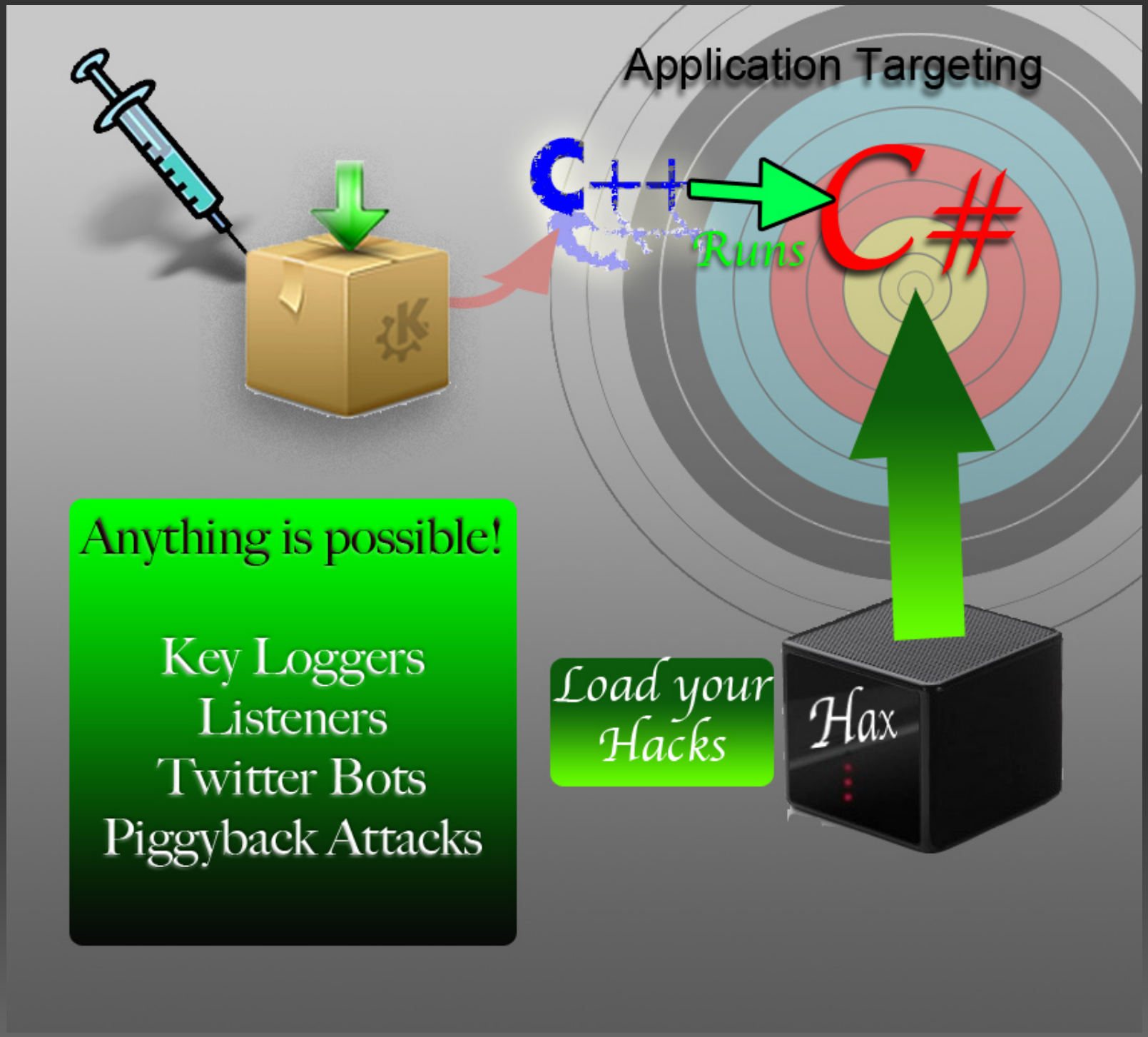
# Reflection

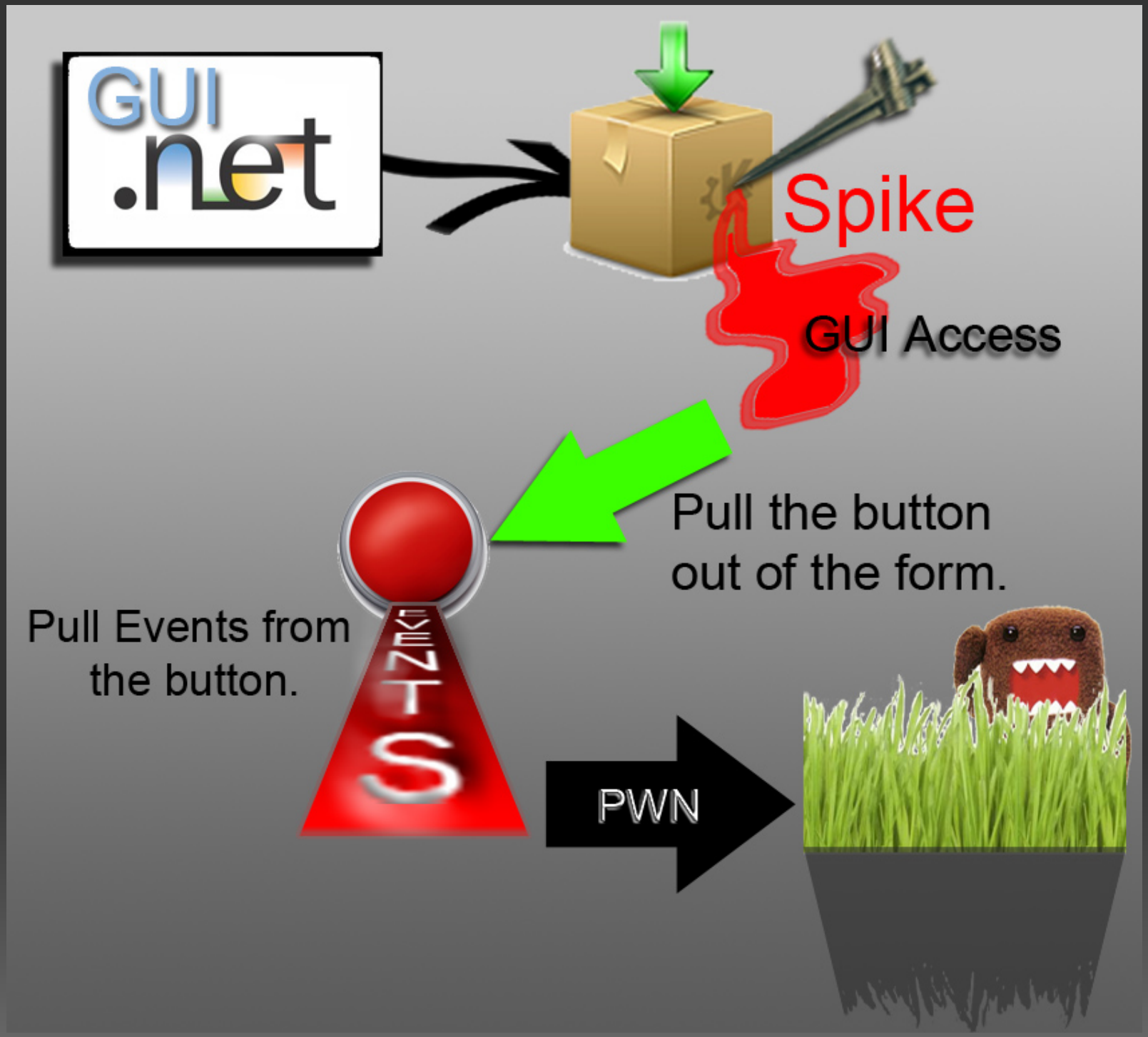
# DEMO

GUI_Spike
LEET a Program

VIDEO OF DEMO HACKS
NOT LIVE DUE TO TIME


APPLICATION
pwn3d!!!


Application Targeting
C++
Runs
C#
Anything is possible!
Key Loggers
Listeners
Twitter Bots
Piggyback Attacks
Load your Hacks
Hax


GUI
.net
Spike
GUI Access
Pull the button
out of the form.
Pull Events from
the button.
PWN

# Live demo

Data Piggyback

SQL

FIN < NULL

# Special Thanks To

Related Works of

James Devlin
www.codingthewheel.com

Sorin Serban
www.sorin.serbans.net/blog

Erez Metula

paper: .NET reverse engineering

& .NET Framework Rootkits

# Thanks to assistance of

Lynn Ackler

Thank you for the mentorship and training in forensics.

Daniel DeFreez

Thank you for the help on research and vulnerability analysis (also the metasploit module) :-)

Andrew Krug

Thank you for the advanced IT support & shinyness.

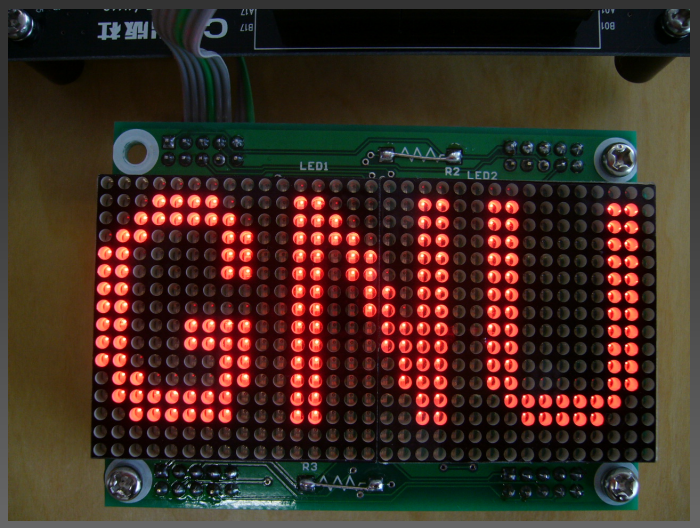
Adam REDACTED

Thanks you for the IT Support; specifically hardware.

# License

DotNetSpike and This Presentation are Licensed Under GNU General Public License - Ver. 3, 29 June 2007

This is an open source presentation presented at Defcon 18 with Tools released at Blackhat 2010



More information at:

http://www.DigitalbodyGuard.com

# How is an attack done

Connect to an Object
Move Objects
Change Objects

Hack Events to change logic

Wrap an Object to replace logic