

The background features a complex, abstract design. It consists of a central, dark, fractal-like structure with intricate, branching patterns. This central element is surrounded by a network of glowing, golden-yellow lines that intersect and form various geometric shapes, including circles and triangles. The overall color palette is dark, with deep blues and blacks, contrasted by the bright, glowing lines and the white text.

# **Visualizing Software Security**

Richard Johnson  
richardi@microsoft.com

# Opening Questions

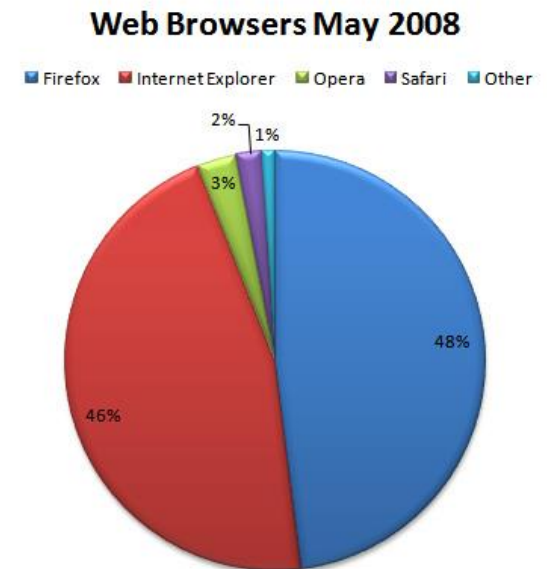
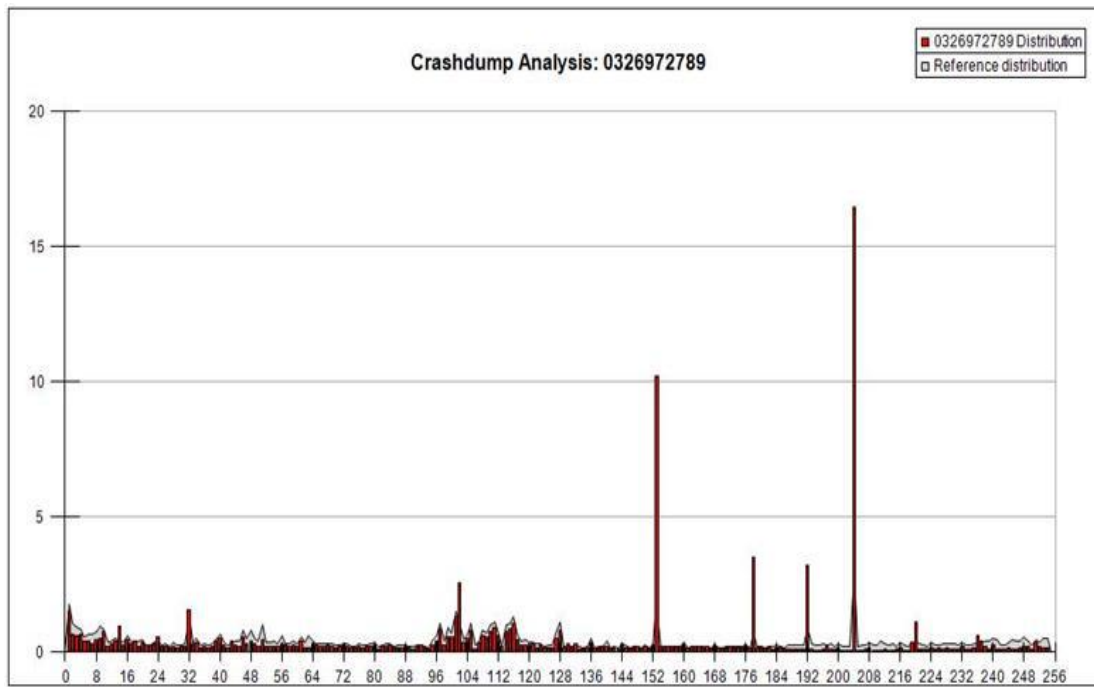
- How can we use the visualization tools we currently have more effectively?
- How can the Software Development Lifecycle benefit from visualizations?
- What is the impact of visualizations on our software security processes?

# Visualization 101

- What is visualization?
  - Information transmission through imagery
- Why is visualization important?
  - Visualizations utilize the mind's most perceptive input mechanism
- What are the challenges in visualization?
  - Create intuitive spatial mappings of non-spatial data
  - Retain clarity while presenting highly dimensional data

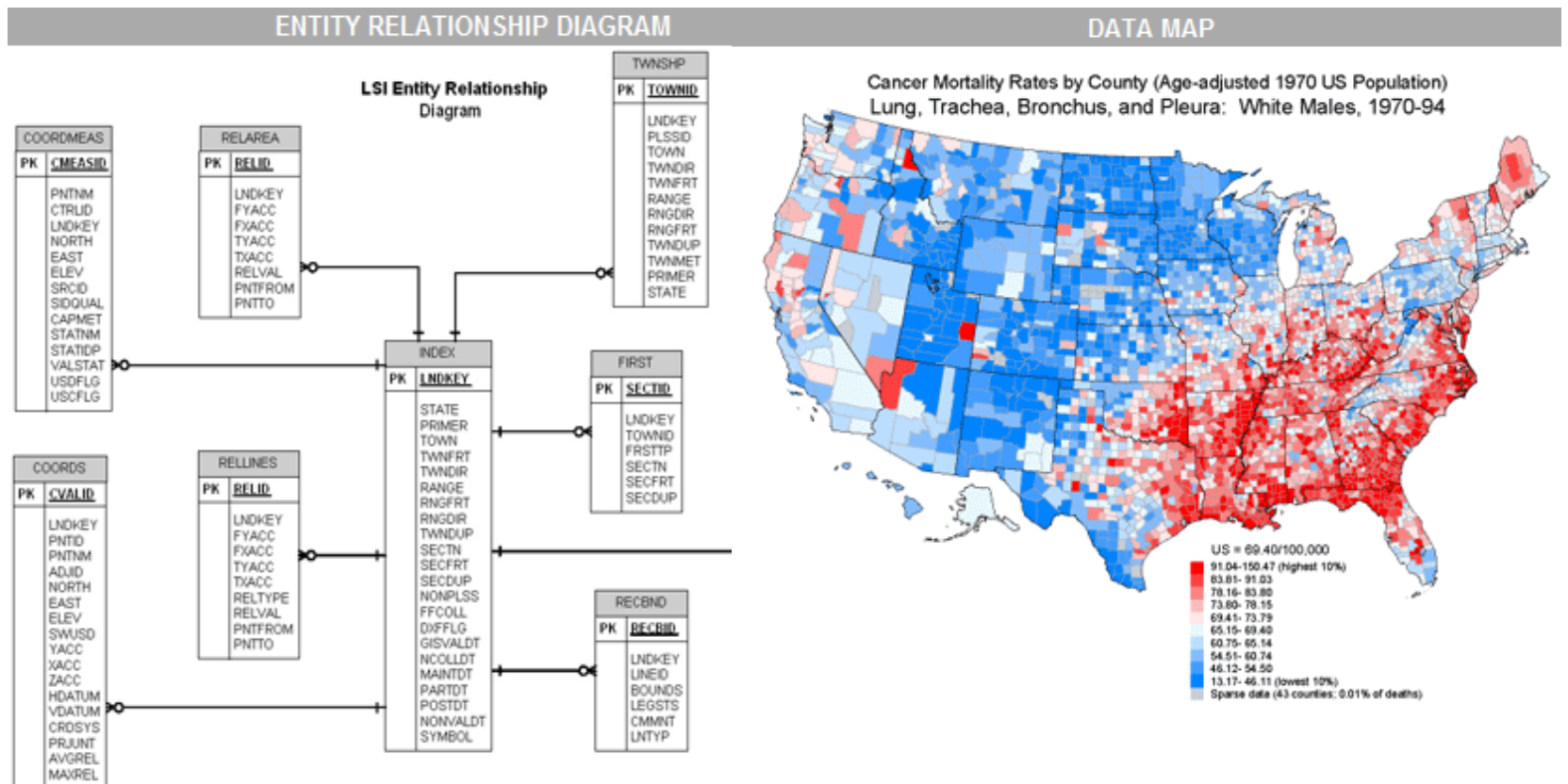
# Visualization Taxonomy

- Data Visualization



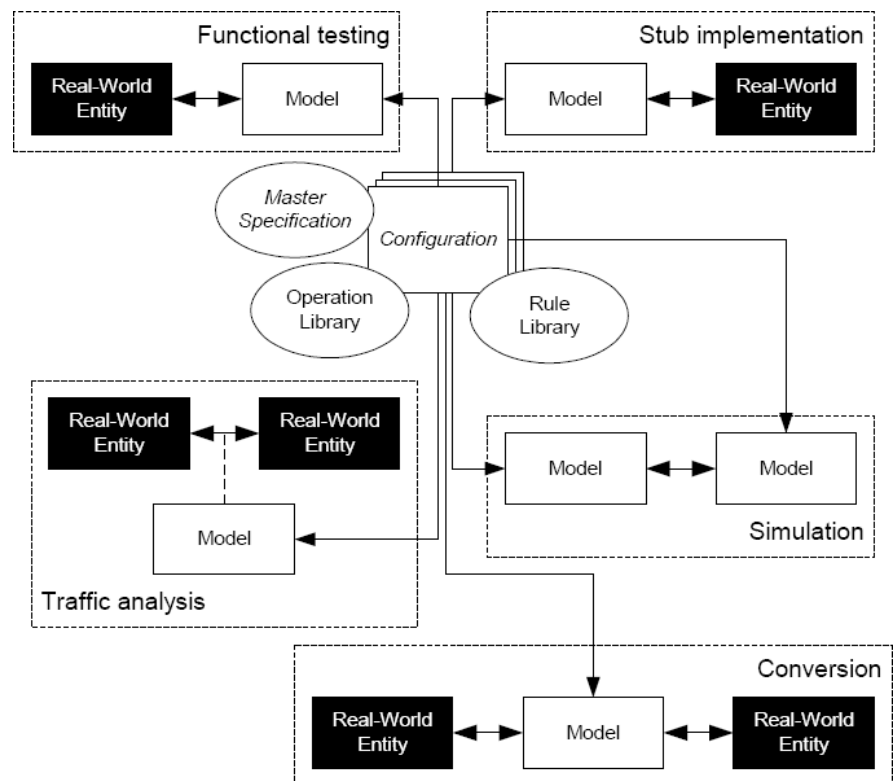
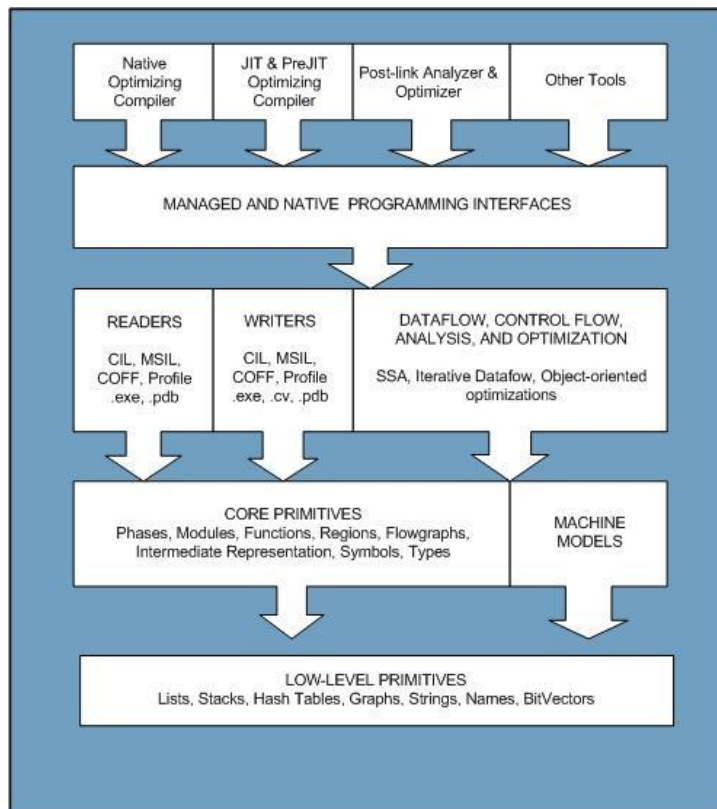
# Visualization Taxonomy

- Information Visualization



# Visualization Taxonomy

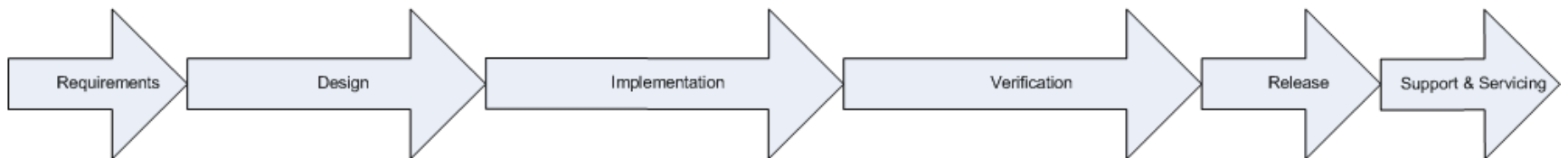
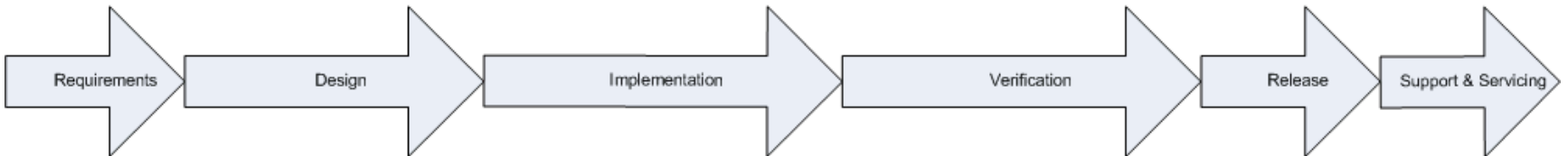
## ■ Concept Visualization





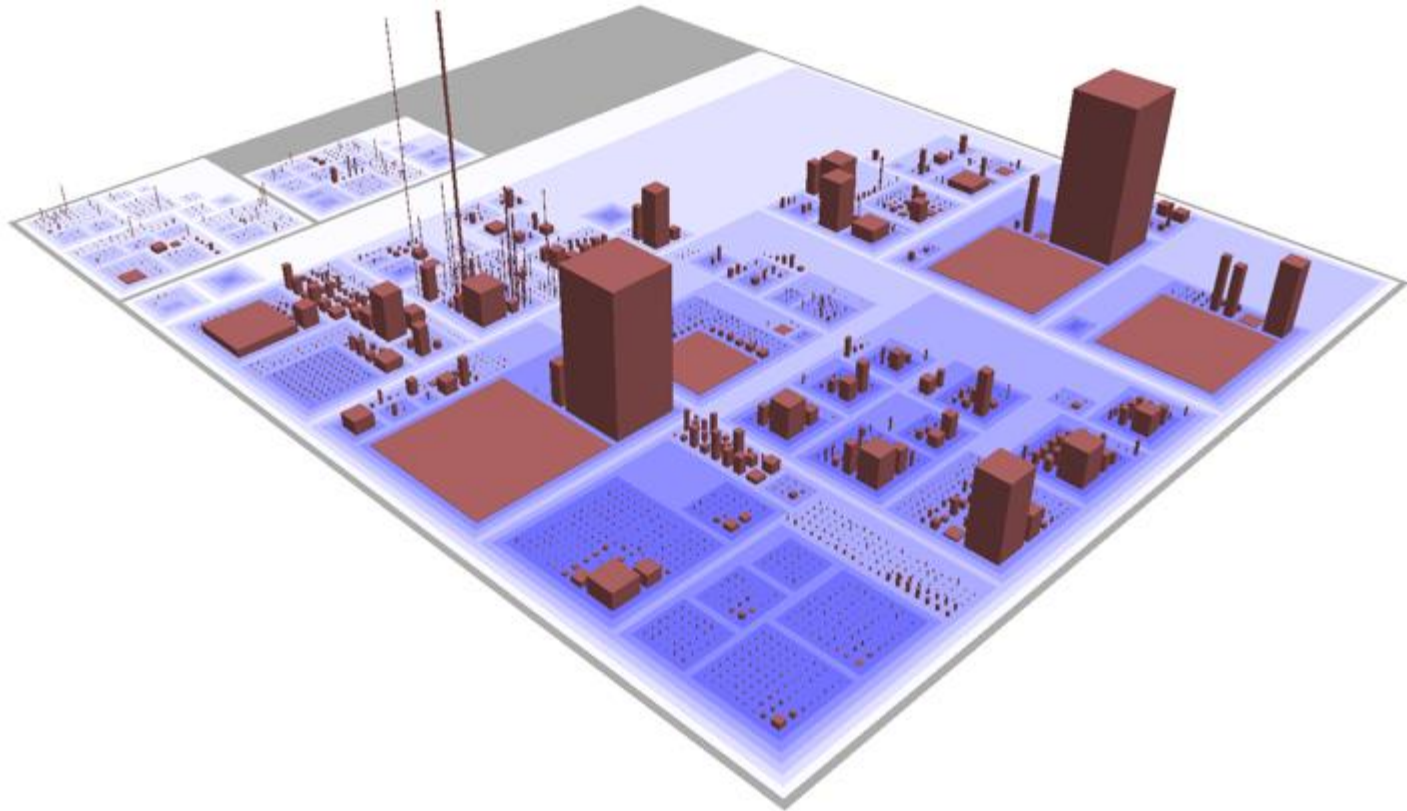
# Visualization Taxonomy

## ■ Strategy Visualization



# Visualization Taxonomy

- Metaphor Visualization





# Software Visualization

- Problem Space
  - Program Visualization
  - Algorithm Visualization
- Sourcing Data
  - Static vs Dynamic data
  - Inaccurate analysis tools
- The goal is always: Reduce Complexity!

# Static Software Properties

- Structural Connectivity
  - Execution & Data Flow
  - Class Hierarchies
- State Machine Models
  - Memory profile
  - Algorithm Complexity
- Revision History
  - Age and authorship
  - Milestones in quality assurance

# Dynamic Software Properties

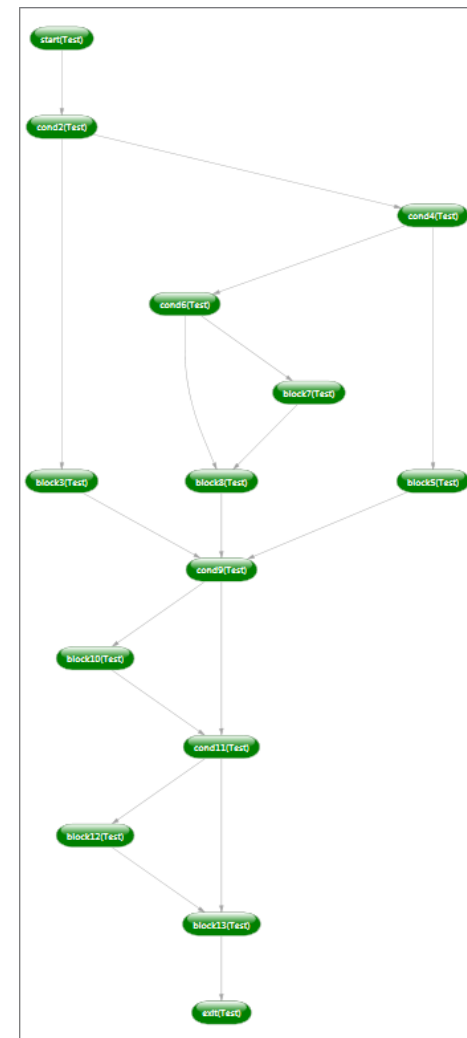
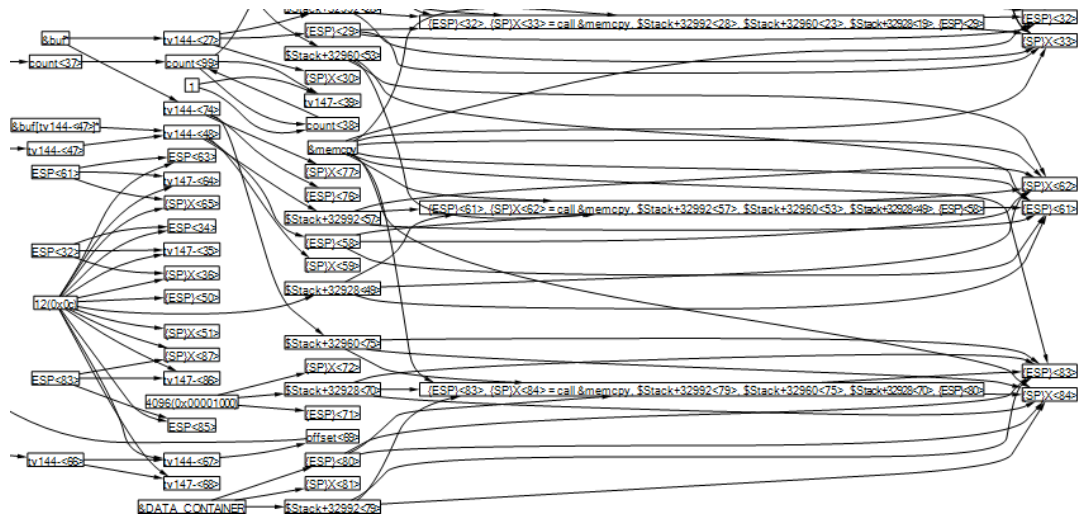
- Execution tracing
  - Code coverage
  - Indirect relationships
  - Dynamic dependencies
- Memory tracing
  - Heap management patterns
  - Object instances
  - Taint propagation
- Environment

# Software Security Properties

- Attack Surface Area
  - Dataflow entry points
  - Privilege boundaries
- Implementation Flaws
  - Arithmetic flaws
  - Comparison flaws
  - Unchecked user input
- Exploitability
  - Execution environment
  - Compiler security
  - Reachability
- History
  - Code age
  - Author credibility

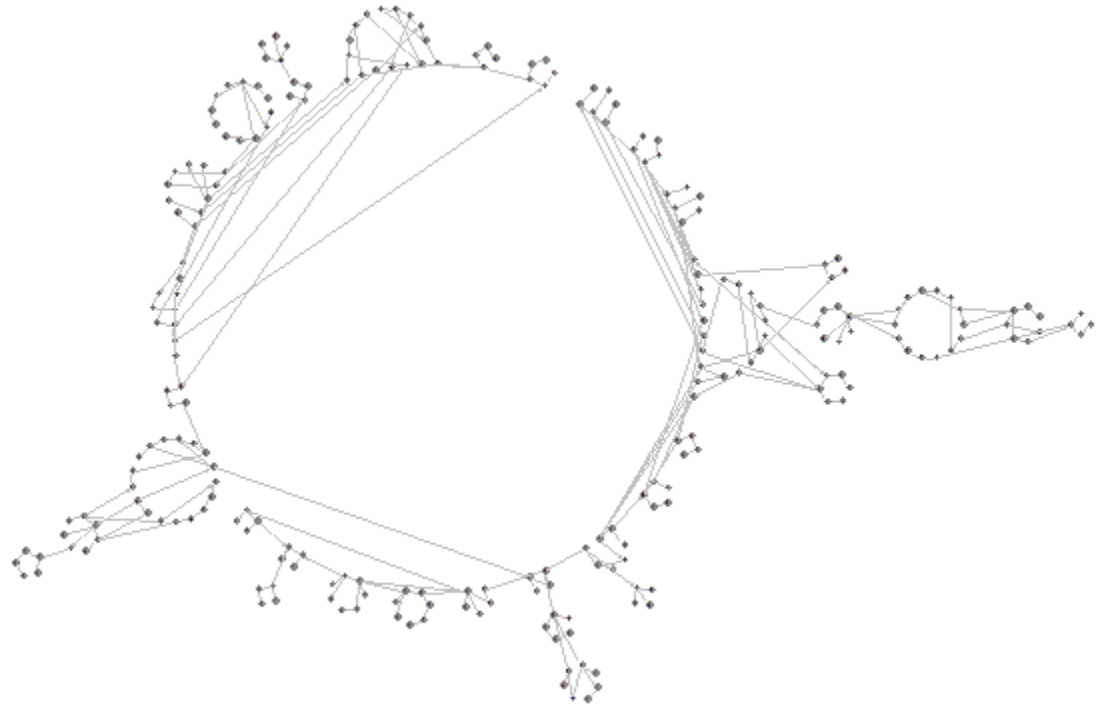
# Graph Visualization

- Hierarchical Layout
  - Layered by order of connectedness
  - Not for highly connected graphs



# Graph Visualization

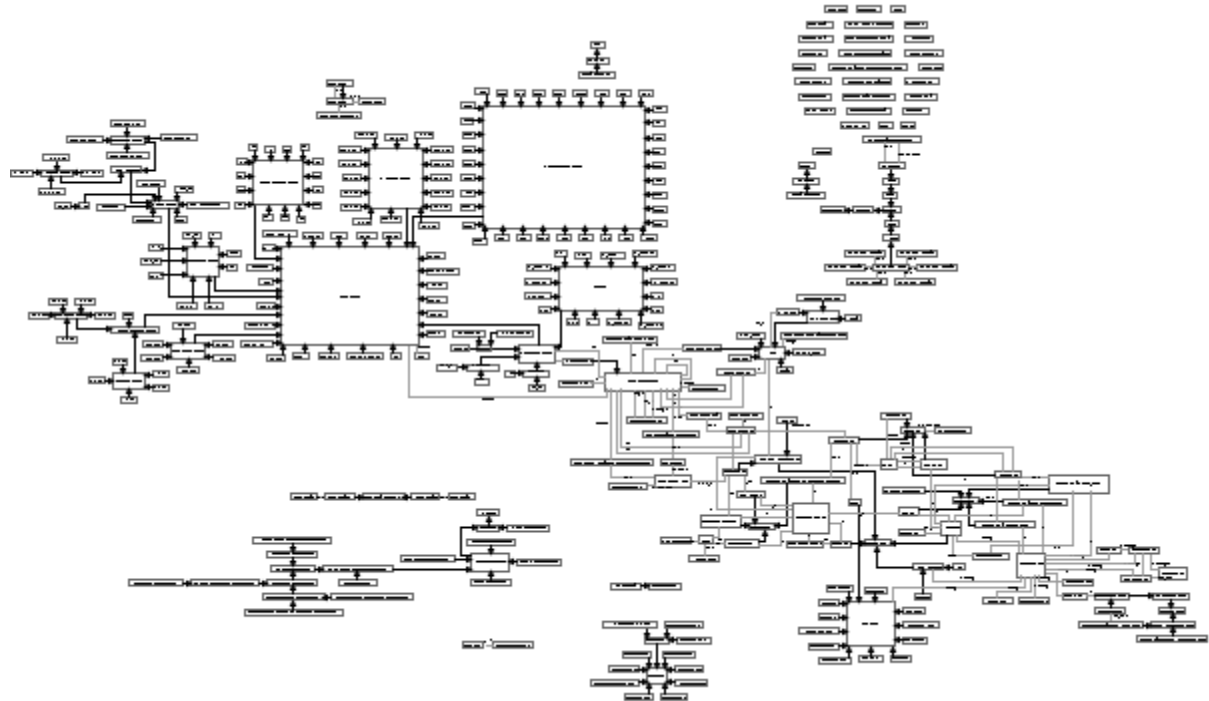
- Circular
  - Nodes aligned on circles
  - Clustering





# Graph Visualization

- Orthogonal
  - Edges aligned on axes
  - Clustering



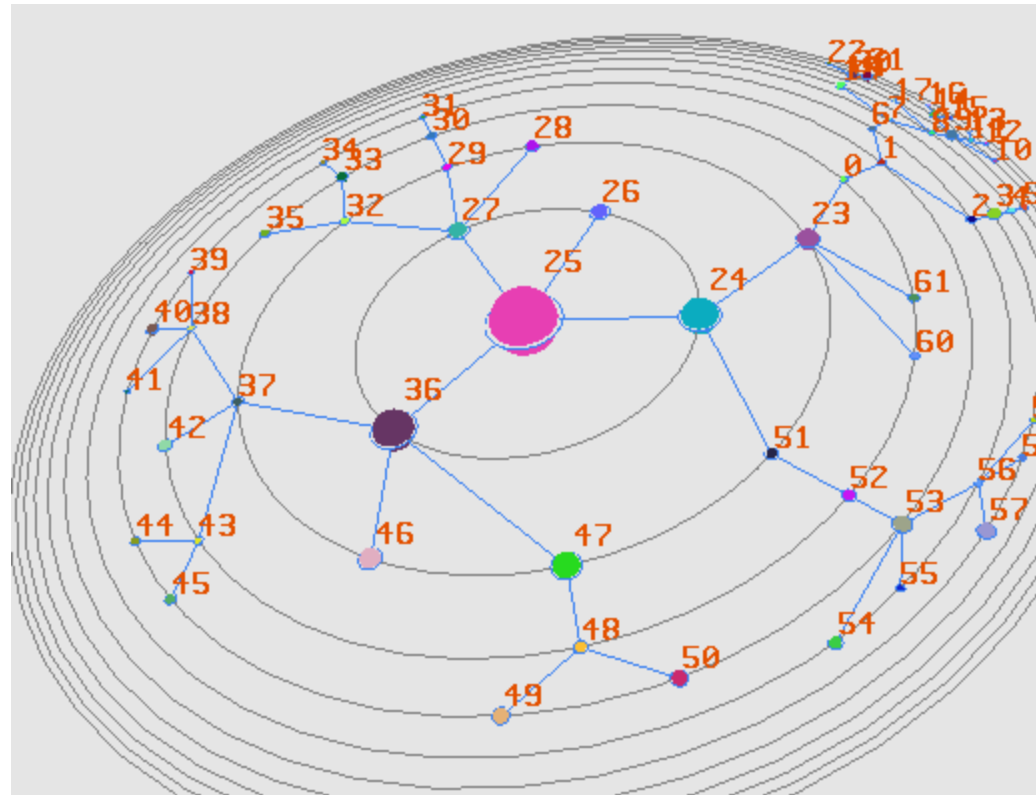
# Graph Visualization

- Force Directed
  - Spring, Magnetic, and Gravitational force
  - Packing



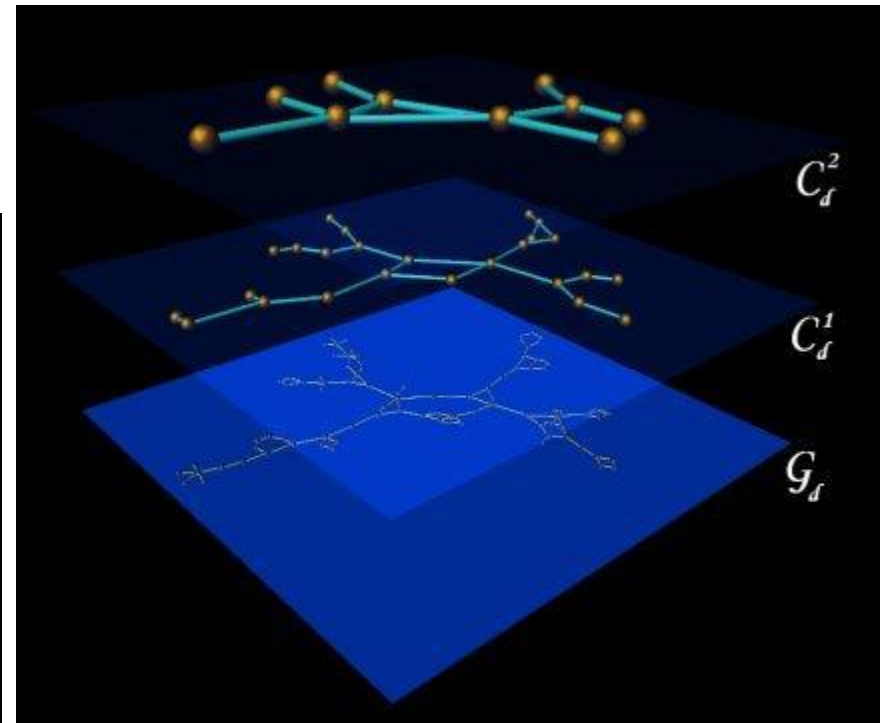
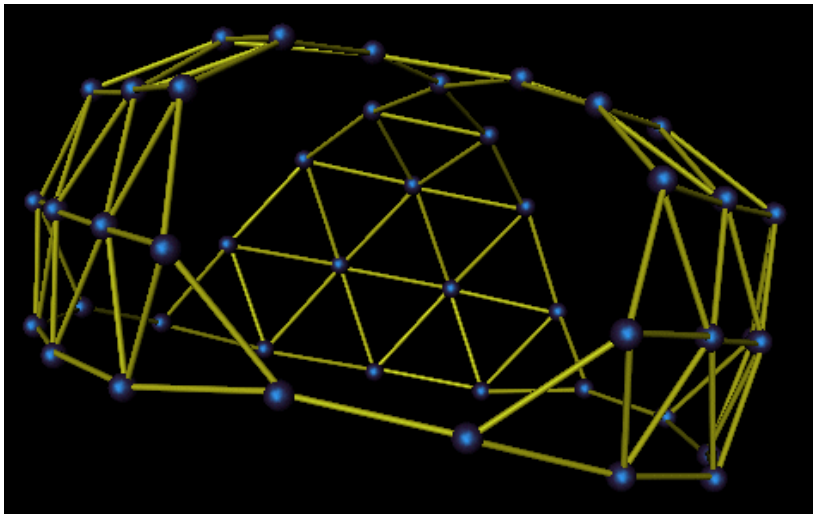
# Improved Graph Visualization

- Hyperbolic Space
  - Clarity on center focus
  - Packing



# Improved Graph Visualization

- Higher Dimensional Space
  - Clarity with high connectivity
  - Multi-level views



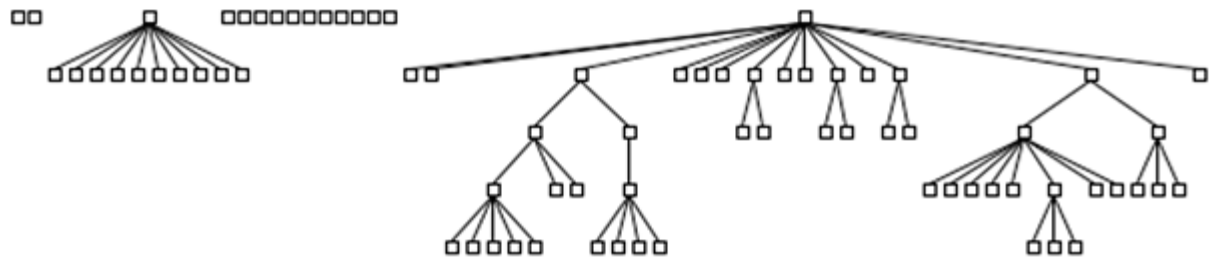
# Visual Attributes

- Nodes

- Spatial coordinates
- Spatial extents
- Color
- Shape

- Edges

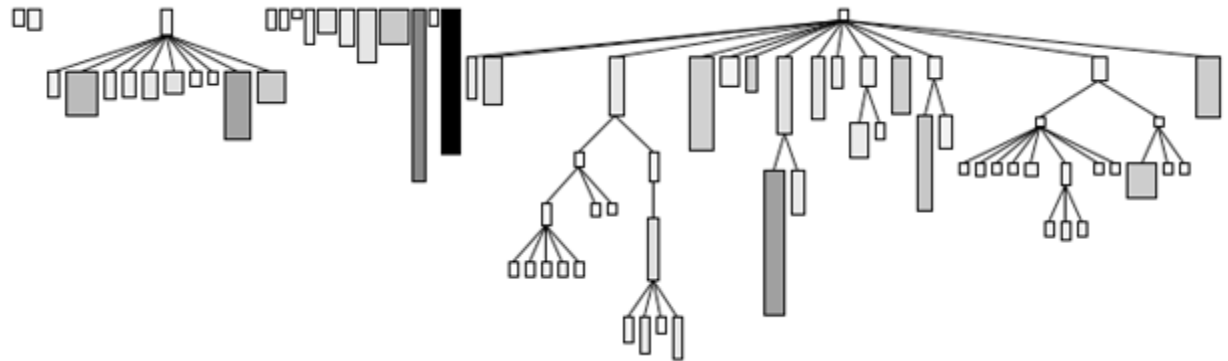
- Color
- Shape
- Width
- Style



# Visual Attributes

- Nodes
  - Spatial coordinates
  - Spatial extents
  - Color
  - Shape

- Edges
  - Color
  - Shape
  - Width
  - Style





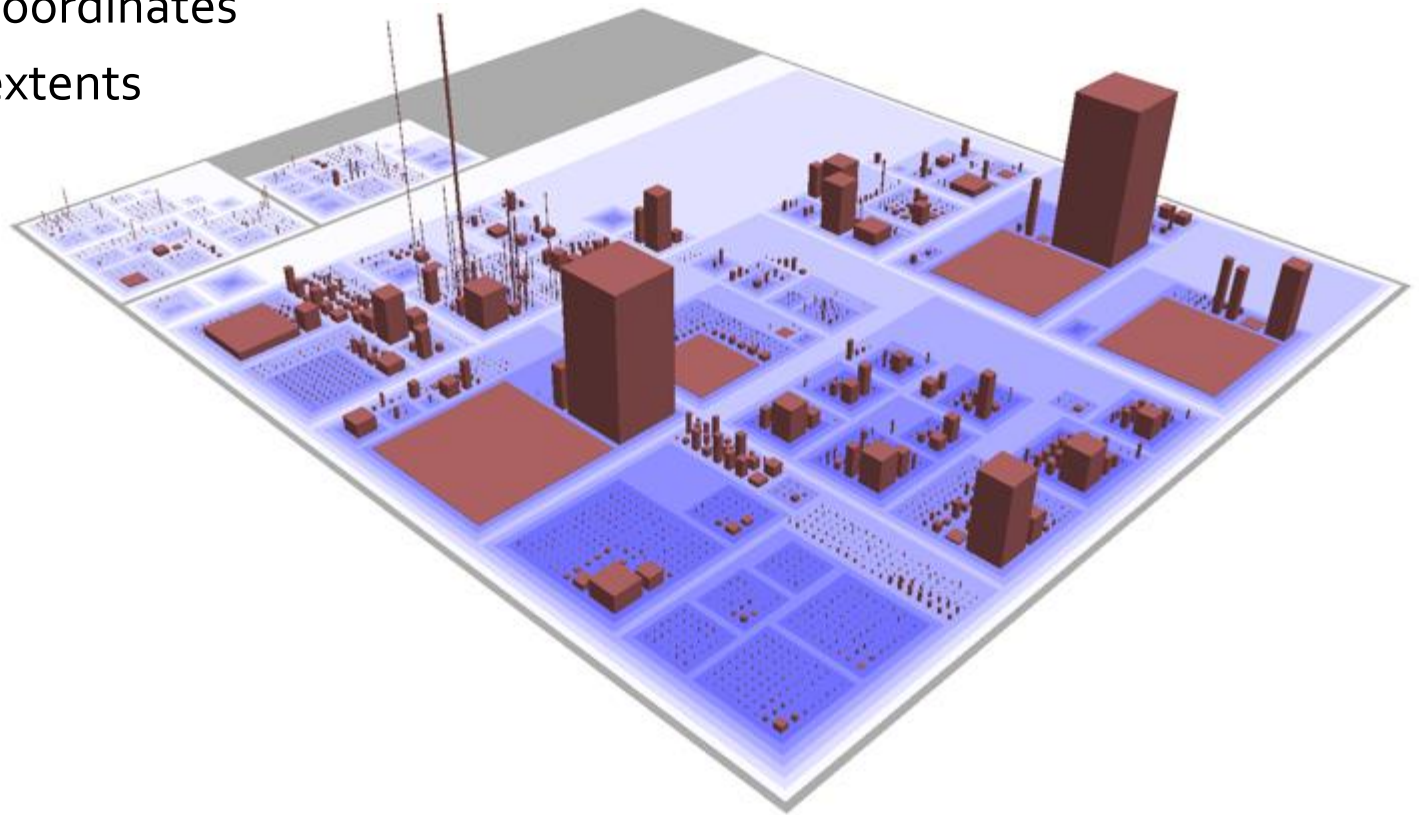
# Visual Attributes

- Nodes

- Spatial coordinates
- Spatial extents
- Color
- Shape

- Edges

- Color
- Shape
- Width
- Style



# Visualizing Software Security

- Observe binary interdependencies



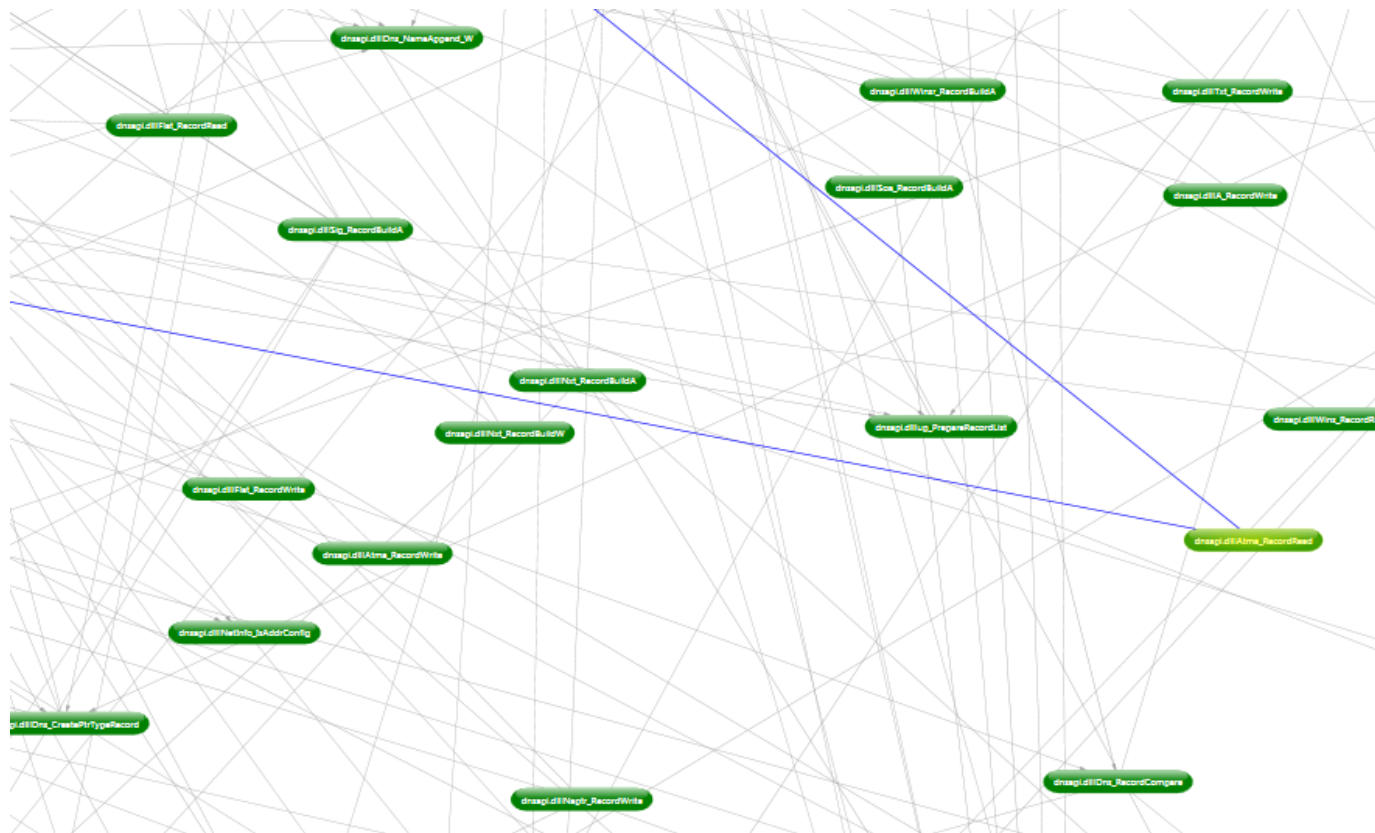
# Visualizing Software Security

- Acquire a method level control flow graph



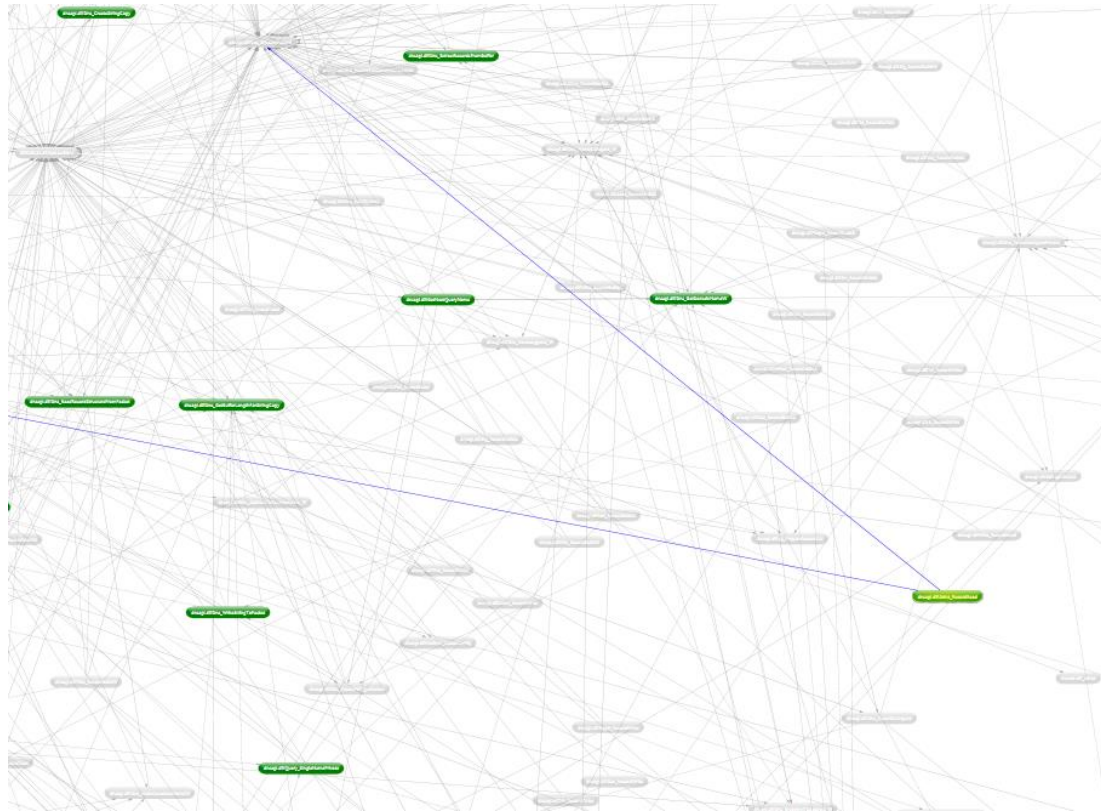
# Visualizing Software Security

- Acquire a method level control flow graph



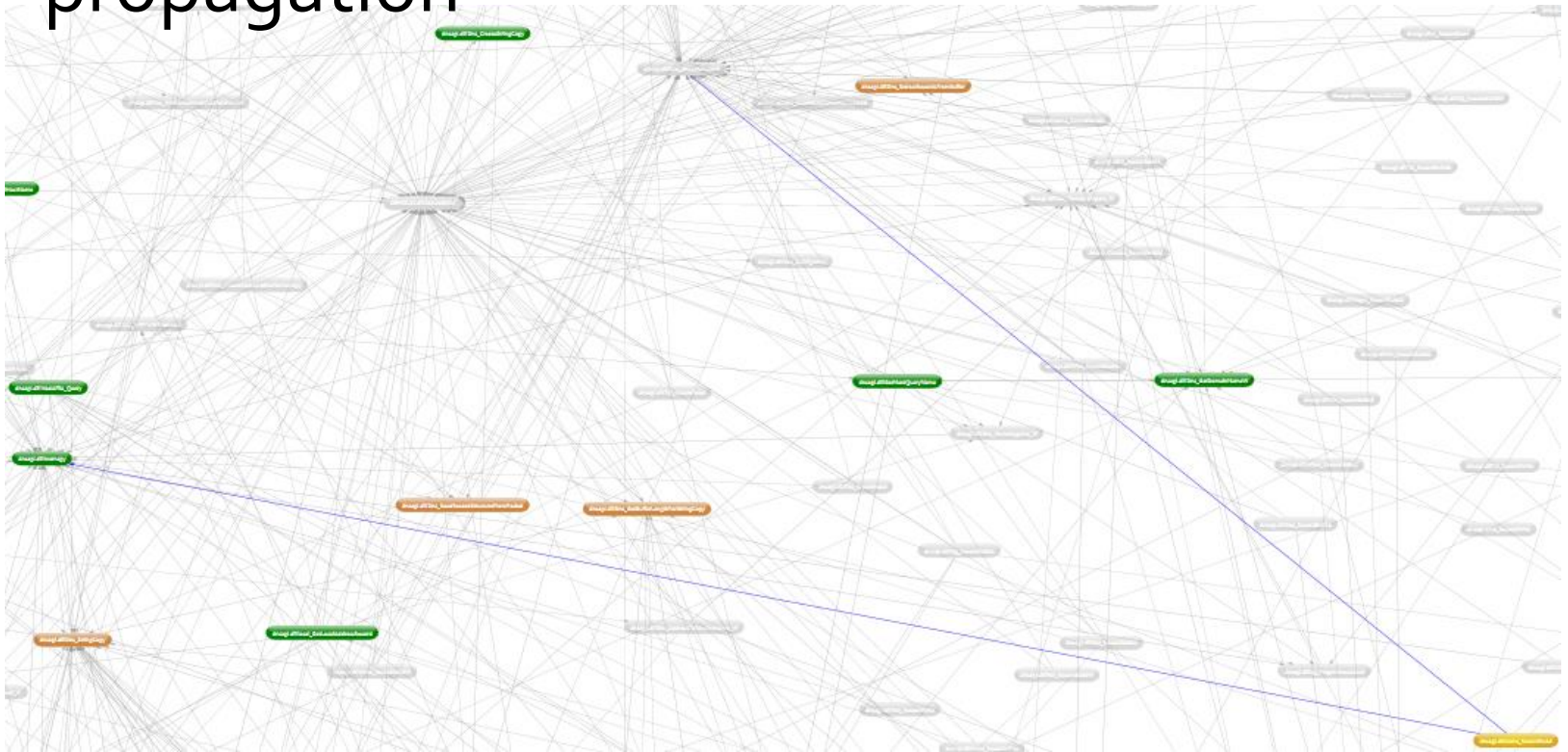
# Visualizing Software Security

- Reduce graph using code coverage data



# Visualizing Software Security

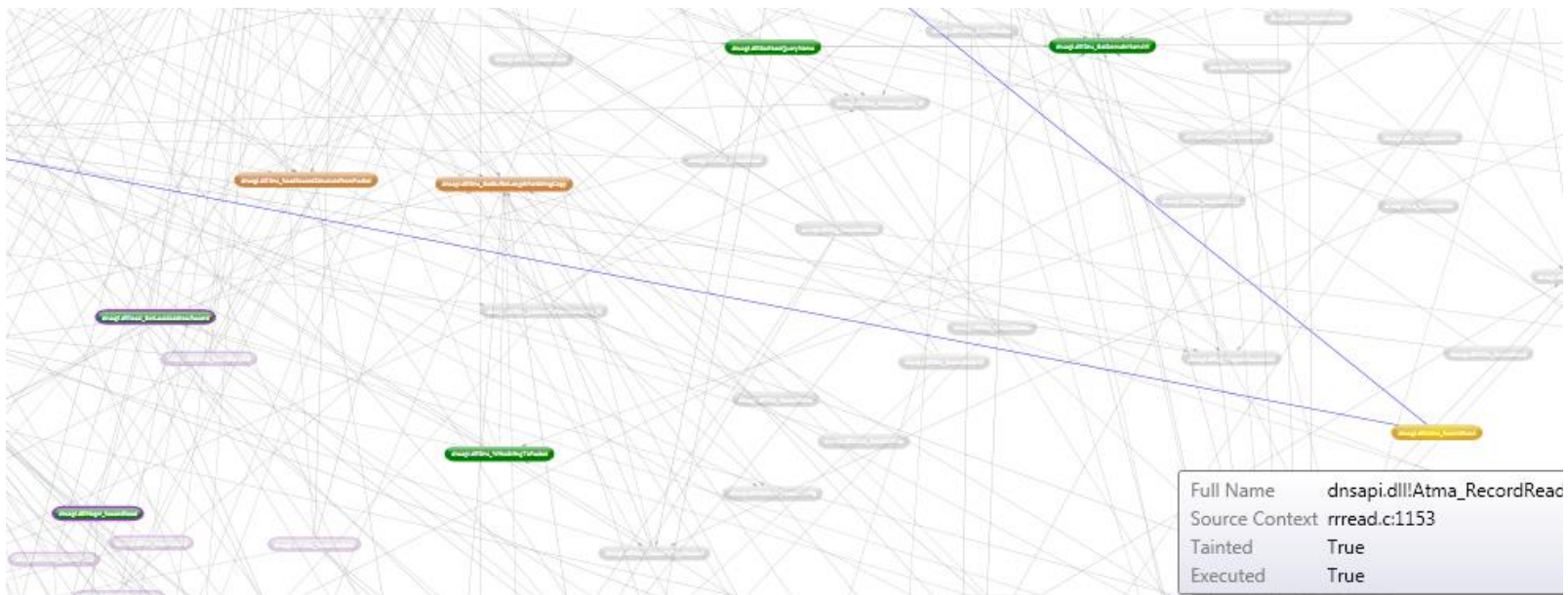
- Trace dataflow dependency to discover taint propagation





# Visualizing Software Security

- Use static analysis plugins to derive security properties such as GS and SafeSEH



# Visualizing Software Security

- Use static analysis plugins to derive security properties such as GS and SafeSEH





# Visualizing Software Security

- Analyze non-covered paths in tainted functions



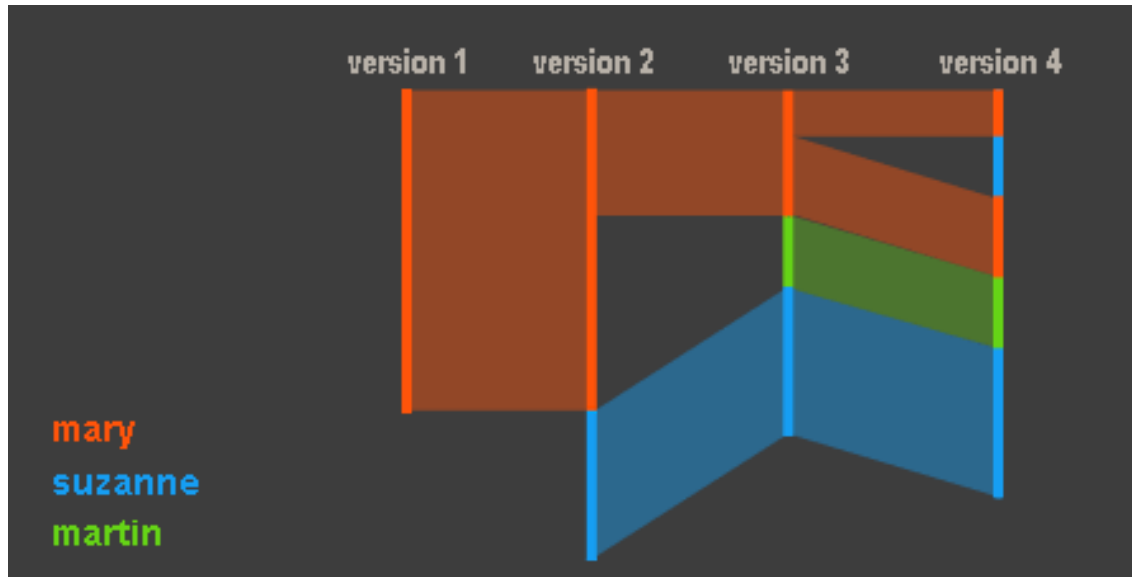
# Visualizing Software Properties

- Examine source code where correlations occur

```
dnsapi.dll!Atma_RecordRead :: rrrread.c:1153
1123 PDNS_RECORD
1124 Atma_RecordRead(
1125     __in_opt          PDNS_RECORD    pRR,
1126     __in              DNS_CHARSET   OutCharSet,
1127     __in              PCHAR         pchStart,
1128     __in_bcount(pchEnd-pchData) PCHAR    pchData,
1129     __in              PCHAR         pchEnd,
1130 )
1131 /*++
1132
1133 Routine Description:
1134     Read ATMA record from wire.
1135
1136 Arguments:
1137     pRR - ptr to record with RR set context
1138
1139     pchStart - start of DNS message
1140
1141     pchData - ptr to RR data field
1142
1143     pchEnd - ptr to byte after data field
1144
1145 Return value:
1146     ptr to new record if successful.
1147     NULL on failure.
1148
1149 --*/
1150 {
1151     PDNS_RECORD precord;
1152     PBYTE pch;
1153     UINT_PTR wireLen = (pchEnd - pchData);
1154
1155     //
1156     // bogus record check
1157     //
1158     if ( wireLen < 2 || wireLen > (DNS_ATMA_MAX_ADDR_LENGTH + sizeof(BYTE)) )
1159     {
1160         return NULL;
1161     }
1162
1163     precord = Dns_AllocateRecord( sizeof( DNS_ATMA_DATA ) +
1164                                 DNS_ATMA_MAX_ADDR_LENGTH );
1165     if ( !precord )
1166     {
1167         return( NULL );
1168     }
1169 }
```

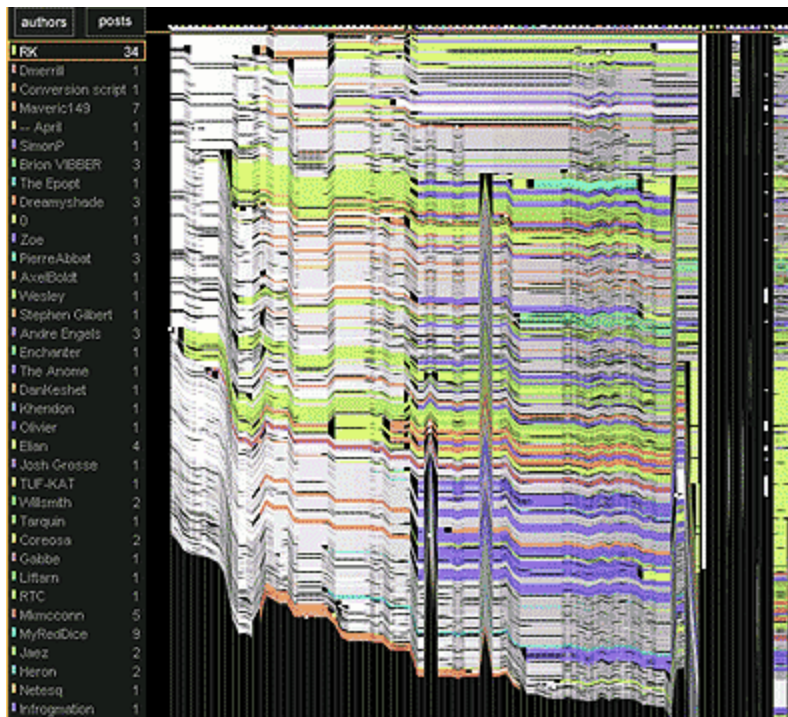
# Beyond Graphs

- Source Code Revision History
  - History Flow



# Beyond Graphs

- Source Code Revision History
  - History Flow



Islam is a monotheistic religion founded in the 600s based on the religious text known as the Quran. According to Islam, the religion was revealed to the Prophet Muhammad when Allah sent an angel to dictate a series of revelations to him, which Muhammad memorized. Muhammad was illiterate, and his followers later wrote down Muhammad's memorized revelations to form the Quran. Muhammad is considered to be the chief and final prophet.

Adherents of Islam are called Muslims (sometimes spelled "Moslem"). In some older English texts they are referred to as "Muhammadans" or "Mohammadan"; however this term is not commonly used because Muslims find it offensive, as this term implies that they worship Muhammad, which they do not.

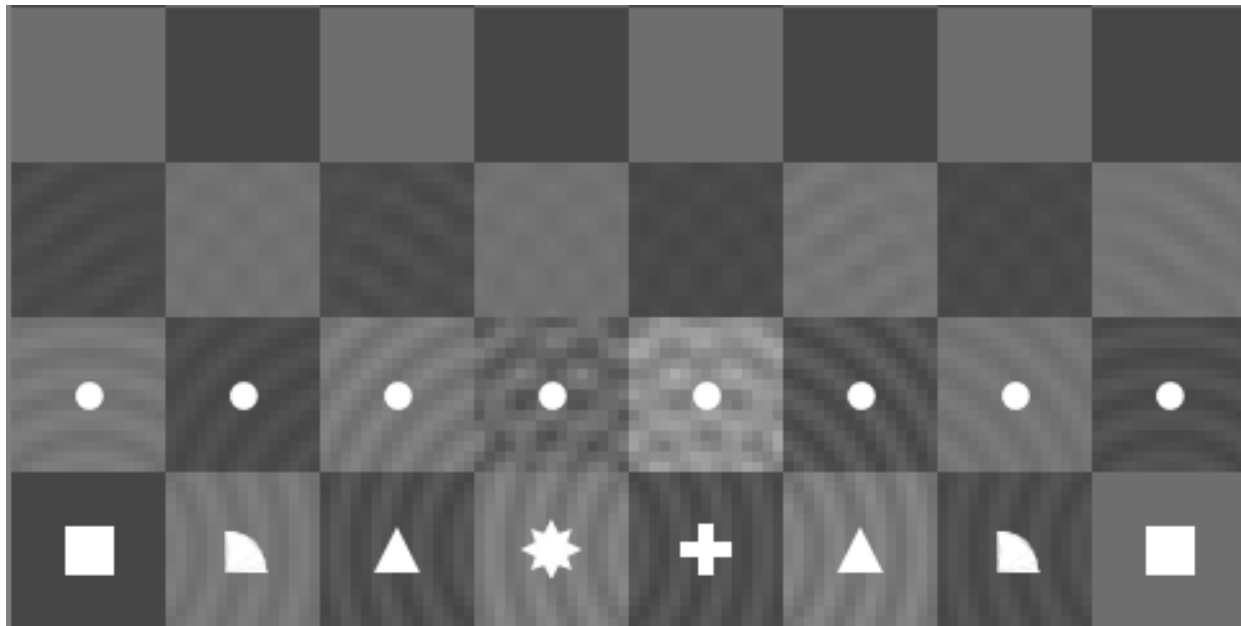
The meaning of the word Islam/Islam is an Arabic word meaning "submission (to Allah)". It also has an etymological relationship to other Arabic words, such as "peace". The word Muslim is derived from Islam and means "one who surrendered" or "submitted (to Allah)".

Teachings of Islam Muslims believe in one God, the God of Adam, Noah, Moses, and Jesus, who are all regarded as prophets or "Messengers" before Muhammad. Muslims believe that Muhammad came to bring the final message of God, the correct path and true knowledge of the afterlife to pagan polytheists and to the Christians and Jews => monotheists who had deviated from the correct path.

For Muslims, the Qur'an answers questions about daily needs, both spiritual and material. It discusses God and God's Names and attributes; believers and their virtues, and the fate of non-believers (kafir); Mary, Jesus, and all the other prophets; and even

# Beyond Graphs

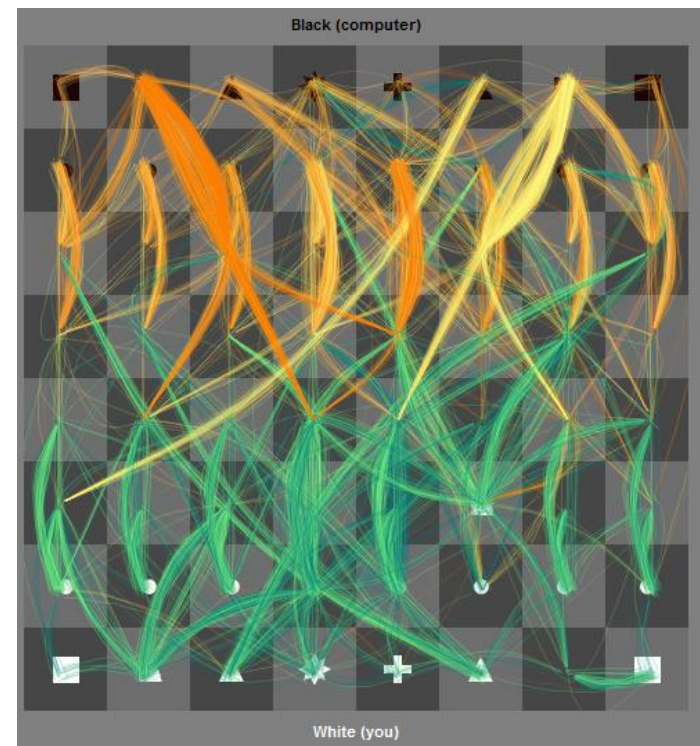
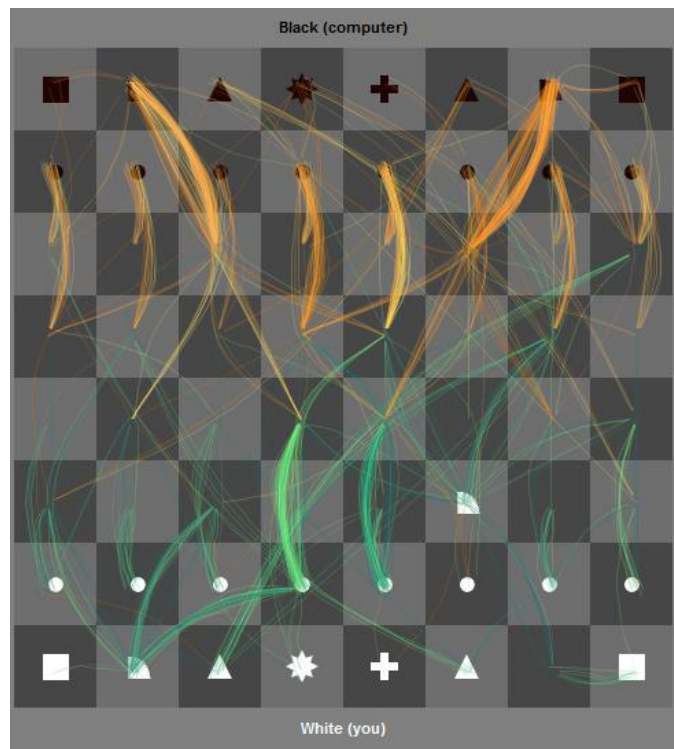
- State Machine Models
  - Thinking Machine





# Beyond Graphs

- State Machine Models
  - Thinking Machine



# Questions?

Richard Johnson  
richardi@microsoft.com





# Thank you!

<http://swiscience>

alias: pandora

Richard Johnson  
richardi@microsoft.com