Visualizing Software Security

Richard Johnson
richardj@microsoft.com

Opening Questions

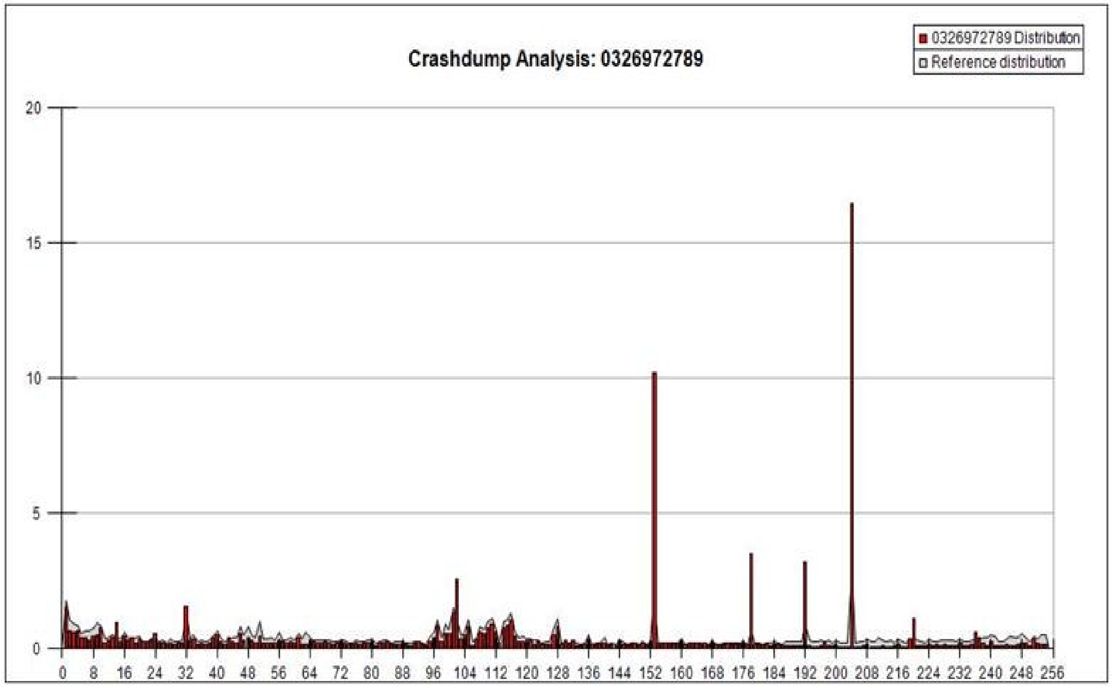
- How can we use the visualization tools we currently have more effectively?
- How can the Software Development Lifecycle benefit from visualizations?
- What is the impact of visualizations on our software security processes?

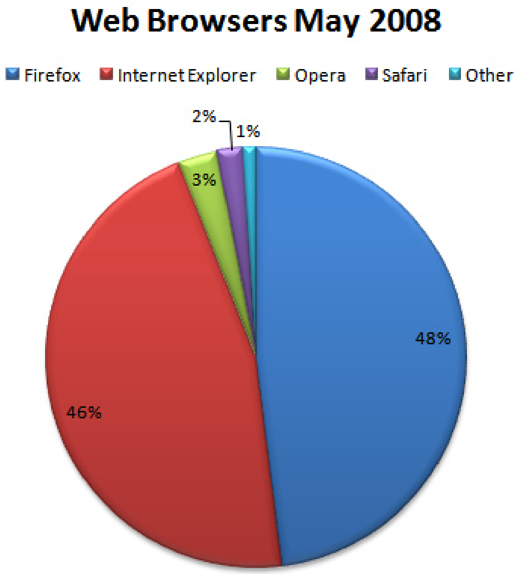
Visualization 101

- 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

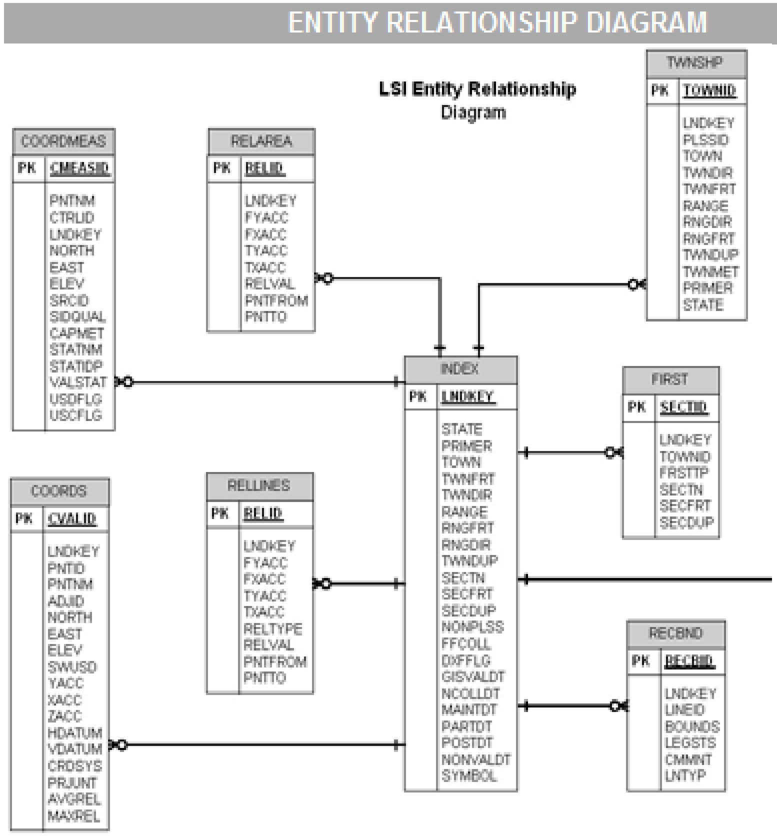
- Data Visualization





Visualization Taxonomy

- Information Visualization



ENTITY RELATIONSHIP DIAGRAM

LSI Entity Relationship
Diagram

COORDMEAS
PK CMEASID
PNTNM
CTRLID
LNDKEY
NORTH
EAST
ELEV
SRCID
SIDQUAL
CAPMET
STATNM
STATIDP
VALSTAT
USDFLG
USCFLG

RELAREA
PK RELID
LNDKEY
FYACC
FXACC
TYACC
TXACC
RELVAL
PNTFROM
PNTTO

TWNSHP
PK TOWNID
LNDKEY
PLSSID
TOWN
TWNDIR
TWNFRT
RANGE
RNGDIR
RNGFRT
TWNDUP
TWNMET
PRIMER
STATE

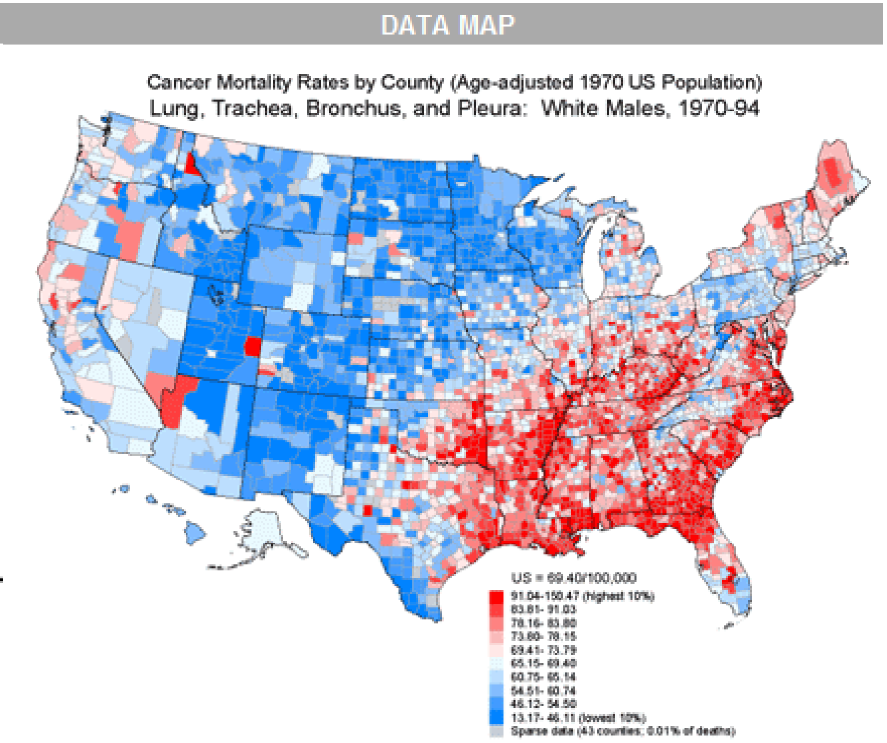
INDEX
PK LNDKEY
STATE
PRIMER
TOWN
TWNFRT
TWNDIR
RANGE
RNGFRT
RNGDIR
TWNDUP
SECTN
SECFRT
SECDUP
NONPLSS
FFCOLL
DXFFLG
GISVALDT
NCOLLDT
MAINTDT
PARTDT
POSTDT
NONVALDT
SYMBOL

FIRST
PK SECTID
LNDKEY
TOWNID
FRSTTP
SECTN
SECFRT
SECDUP

COORDS
PK CVALID
LNDKEY
PNTID
PNTNM
ADJID
NORTH
EAST
ELEV
SWUSD
YACC
XACC
ZACC
HDATUM
VDATUM
CRDSYS
PRJUNT
AVGREL
MAXREL

RELLINES
PK RELID
LNDKEY
FYACC
FXACC
TYACC
TXACC
RELTYPE
RELVAL
PNTFROM
PNTTO

RECBND
PK RECBID
LNDKEY
LINEID
BOUNDS
LEGSTS
CMMNT
LNTYP

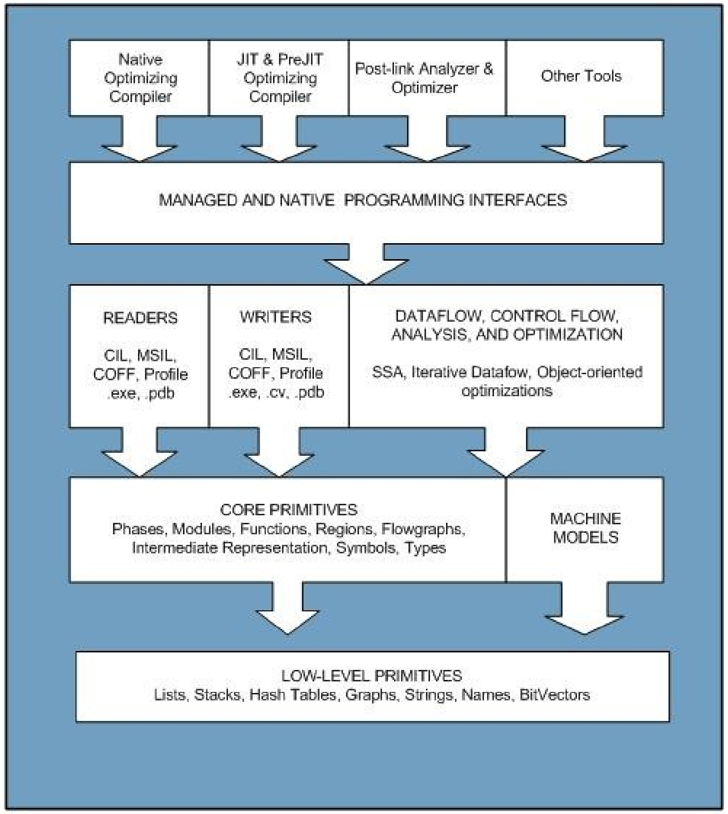


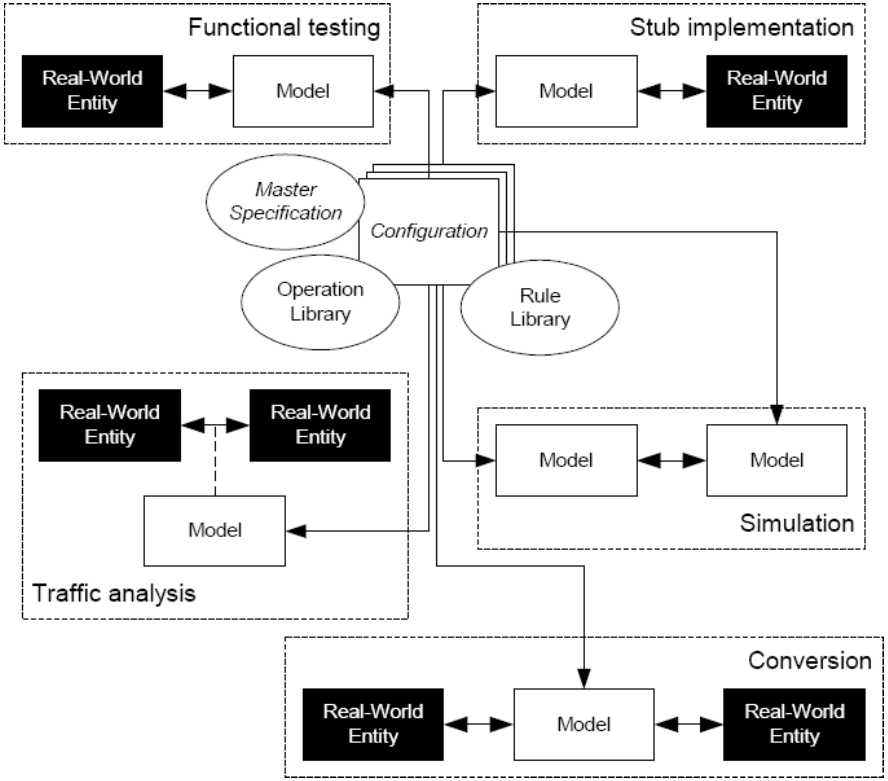
DATA MAP

Cancer Mortality Rates by County (Age-adjusted 1970 US Population)
Lung, Trachea, Bronchus, and Pleura: White Males, 1970-94

Visualization Taxonomy

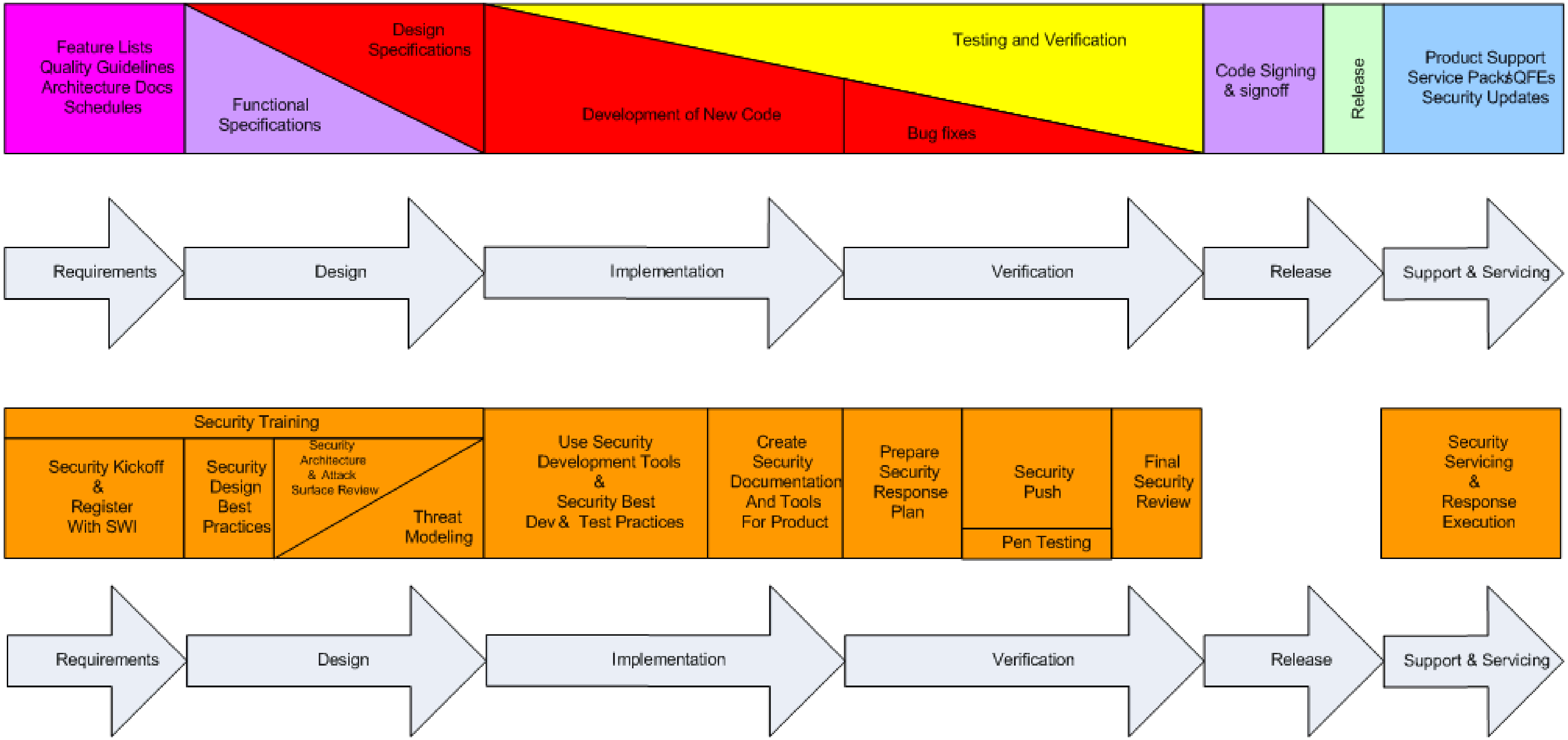
- Concept Visualization


Native Optimizing Compiler
JIT & PreJIT Optimizing Compiler
Post-link Analyzer & Optimizer
Other Tools
MANAGED AND NATIVE PROGRAMMING INTERFACES
READERS
CIL, MSIL, COFF, Profile .exe, .pdb
WRITERS
CIL, MSIL, COFF, Profile .exe, .cv, .pdb
DATAFLOW, CONTROL FLOW, ANALYSIS, AND OPTIMIZATION
SSA, Iterative Datafow, Object-oriented optimizations
CORE PRIMITIVES
Phases, Modules, Functions, Regions, Flowgraphs, Intermediate Representation, Symbols, Types
MACHINE MODELS
LOW-LEVEL PRIMITIVES
Lists, Stacks, Hash Tables, Graphs, Strings, Names, BitVectors



Visualization Taxonomy

- Strategy Visualization



Feature Lists
Quality Guidelines
Architecture Docs
Schedules

Functional
Specifications

Design
Specifications

Development of New Code

Bug fixes

Testing and Verification

Code Signing
& signoff

Release

Product Support
Service Packs/QFEs
Security Updates

Requirements

Design

Implementation

Verification

Release

Support & Servicing

Security Training

Security Kickoff
&
Register
With SWI

Security
Design
Best
Practices

Security
Architecture
& Attack
Surface Review

Threat
Modeling

Use Security
Development Tools
&
Security Best
Dev & Test Practices

Create
Security
Documentation
And Tools
For Product

Prepare
Security
Response
Plan

Security
Push

Pen Testing

Final
Security
Review

Security
Servicing
&
Response
Execution

Requirements

Design

Implementation

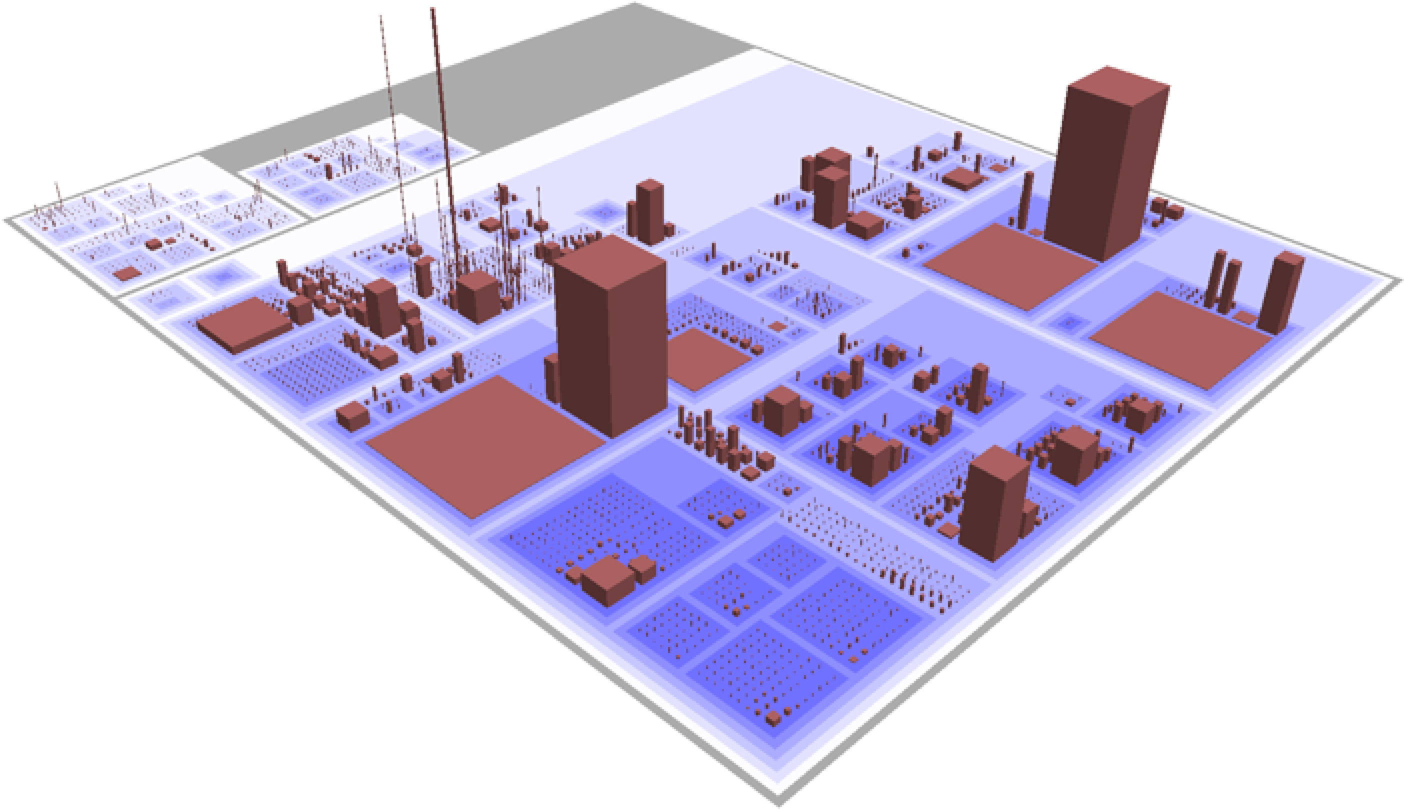
Verification

Release

Support & Servicing

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

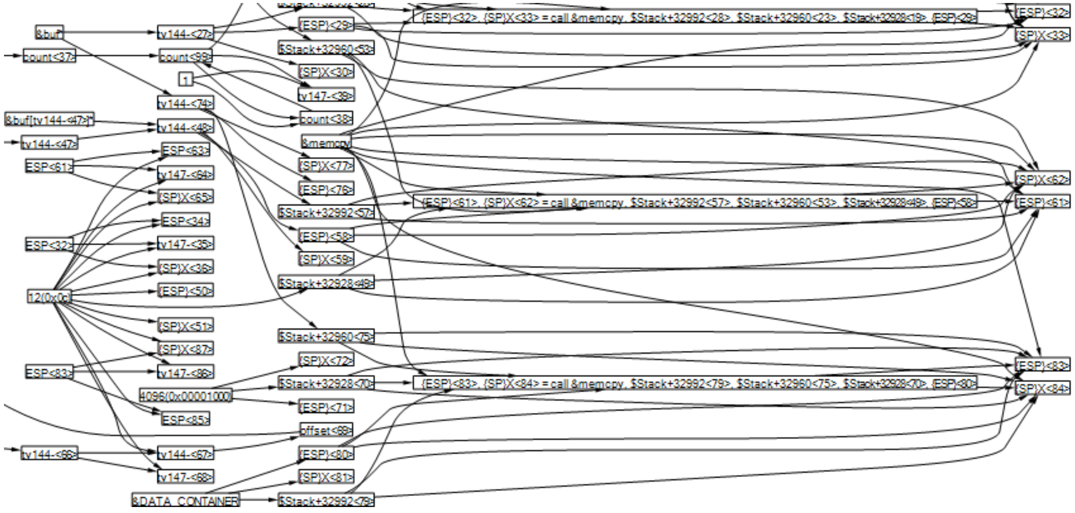
- Execution tracing
  - Code coverage
  - Indirect relationships
  - Dynamic dependencies
- Memory tracing
  - Heap management patterns
  - Object instances
  - Taint propagation
- Environment

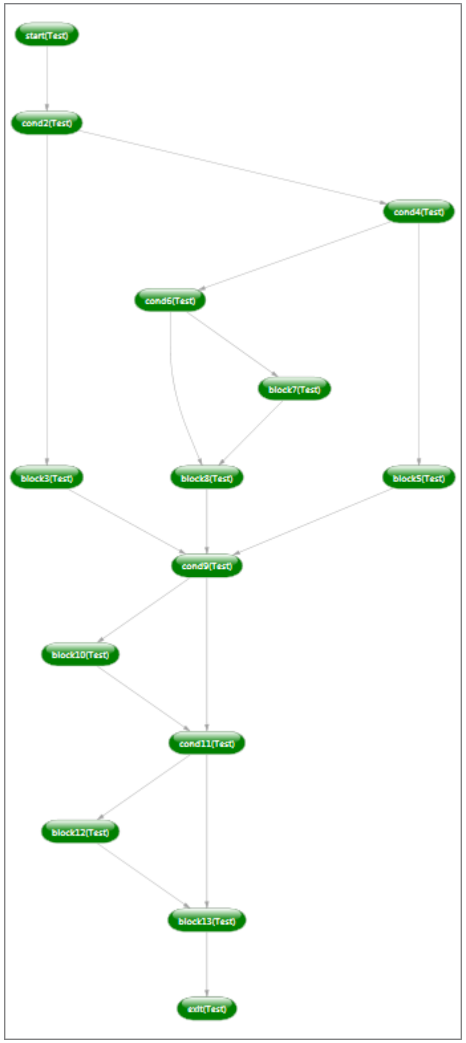
Software Security Properties

- 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

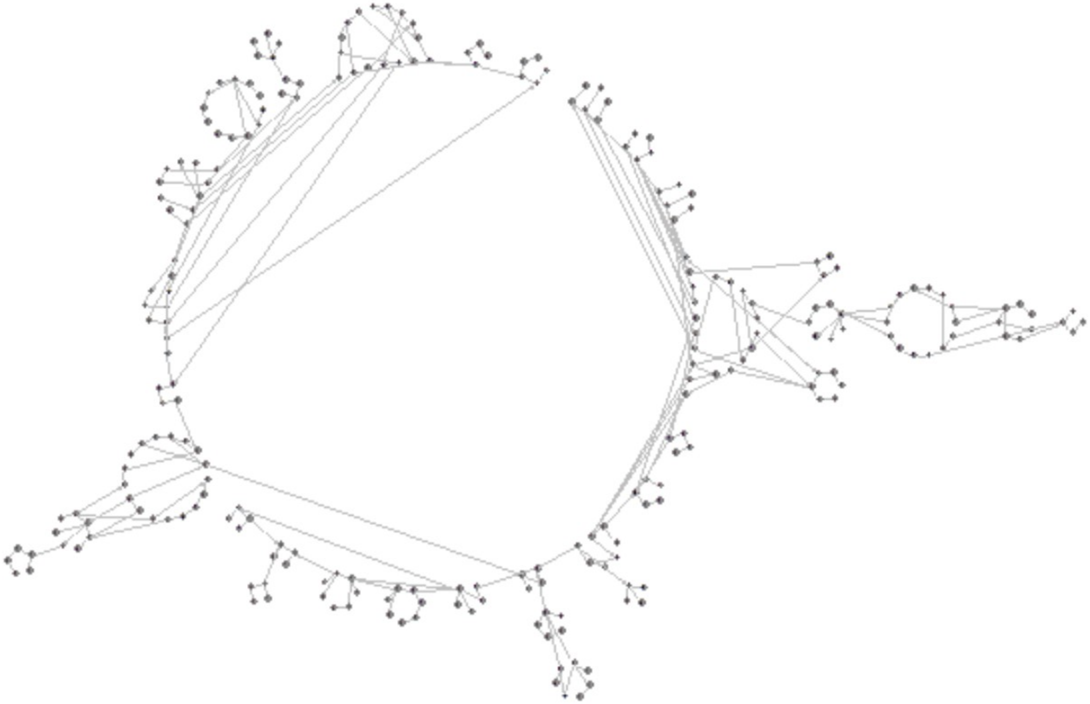
- Hierarchical Layout
  - Layered by order of connectedness
  - Not for highly connected graphs





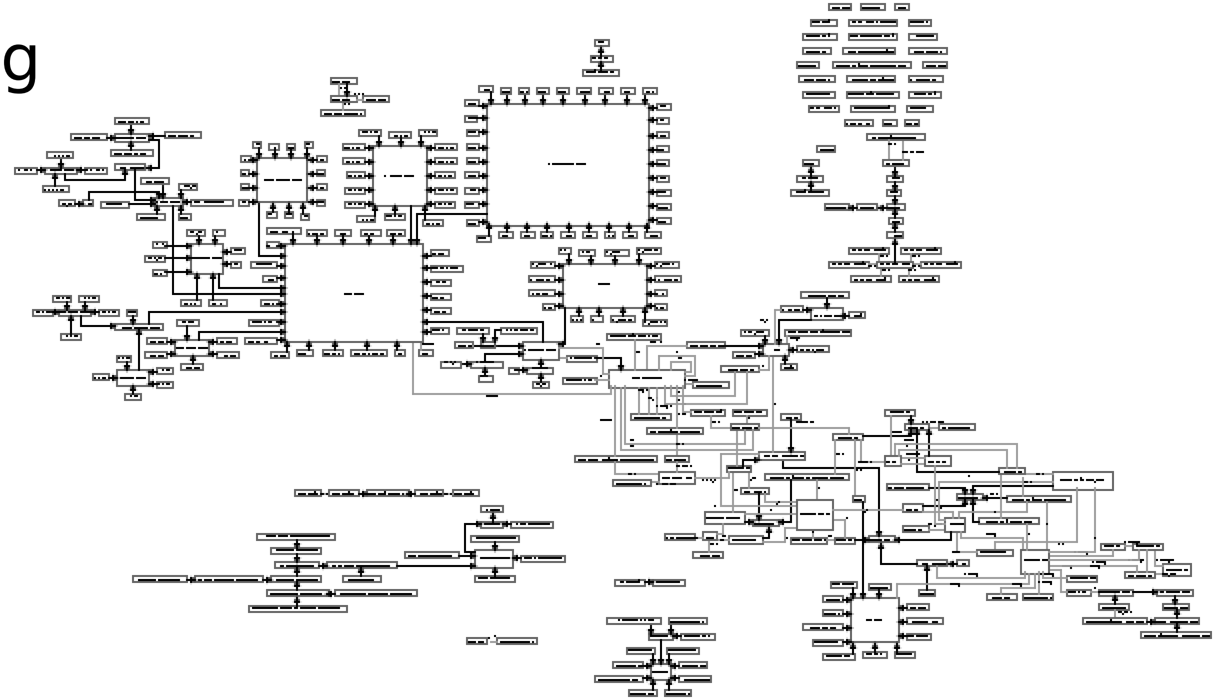
Graph Visualization

- Circular
  - Nodes aligned on circles
  - Clustering



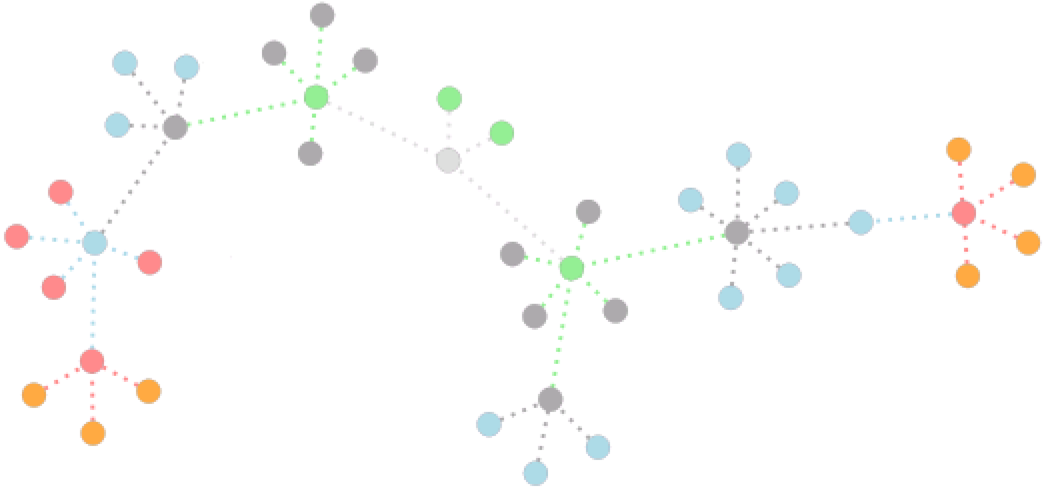
Graph Visualization

- Orthogonal
  - Edges aligned on axes
  - Clustering



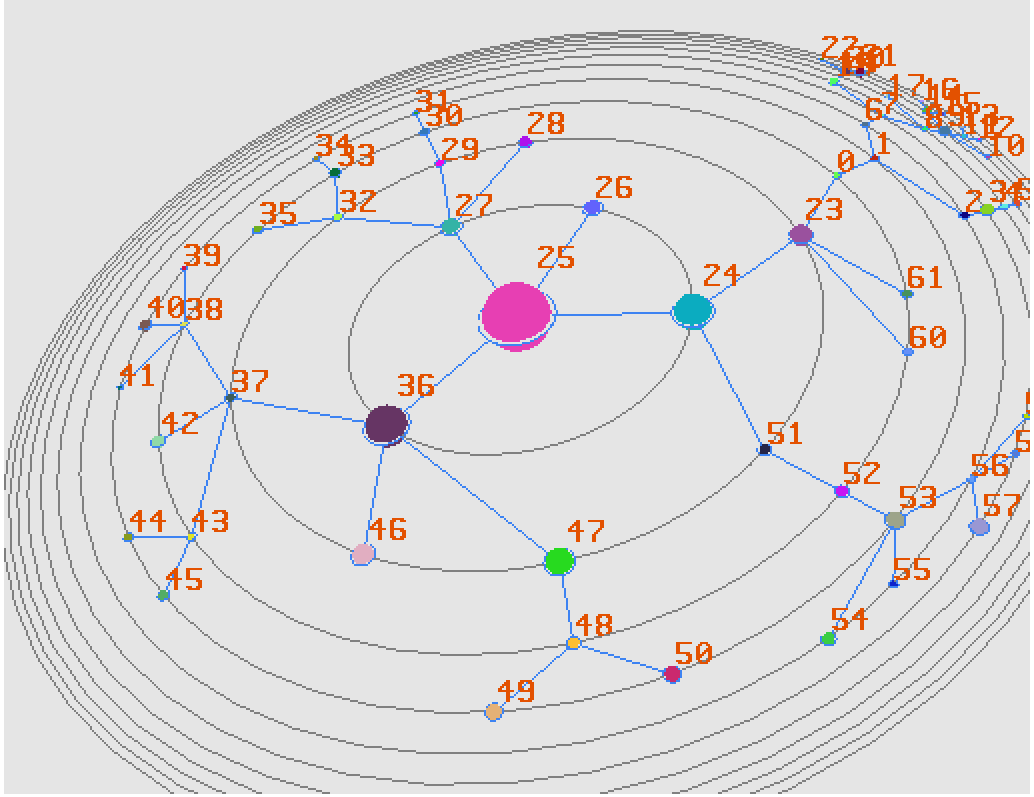
Graph Visualization

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



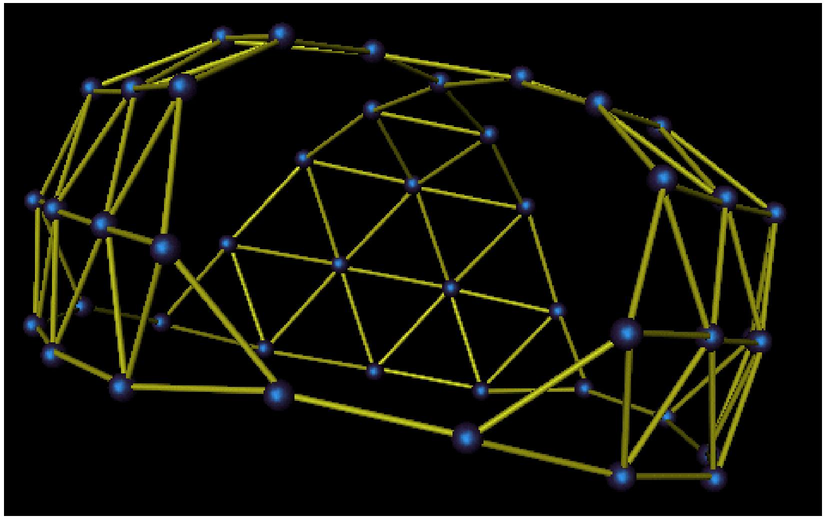
Improved Graph Visualization

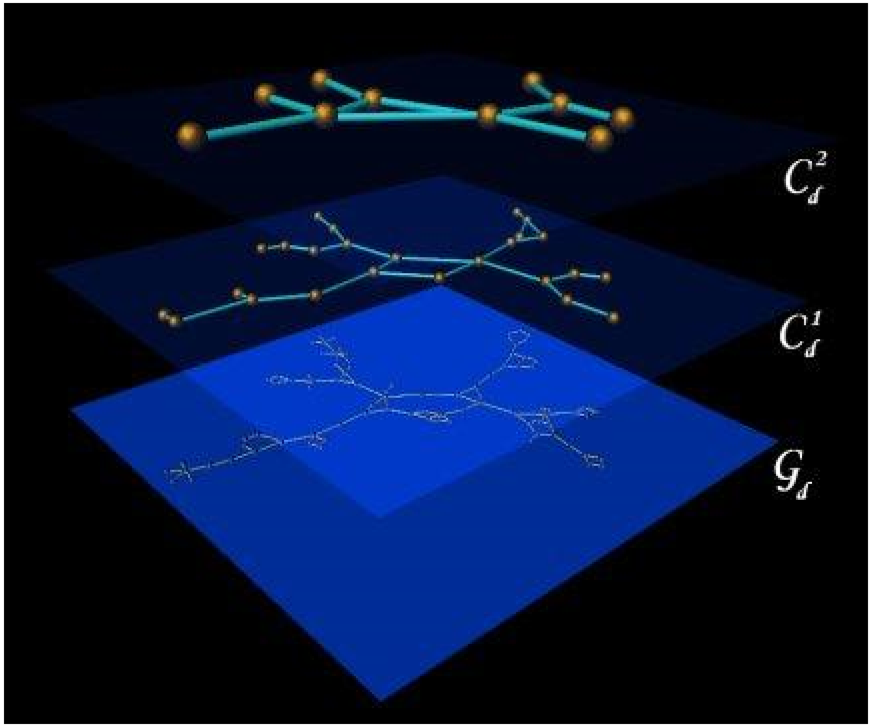
- Hyperbolic Space
  - Clarity on center focus
  - Packing



Improved Graph Visualization

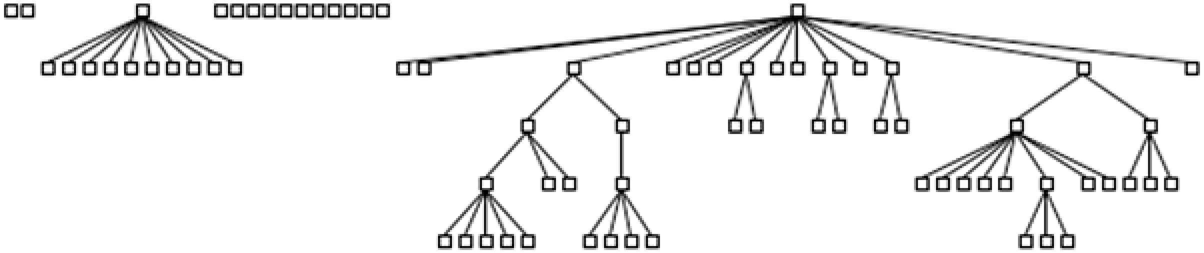
- Higher Dimensional Space
  - Clarity with high connectivity
  - Multi-level views





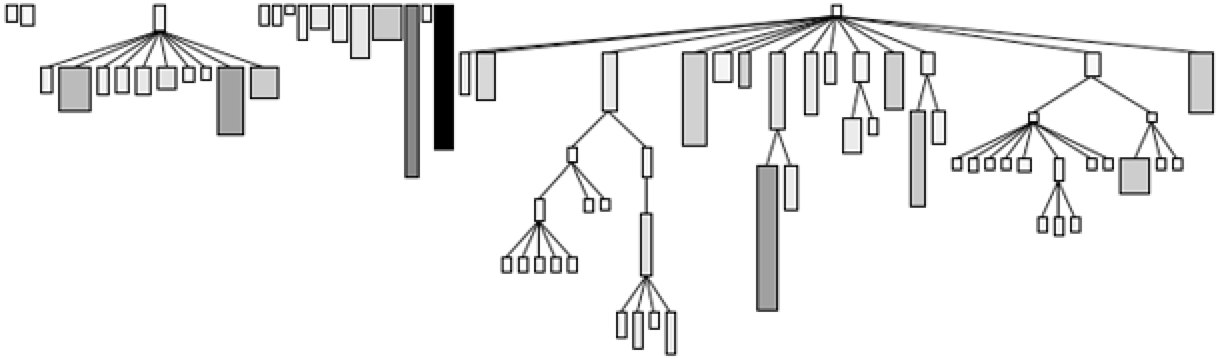
Visual Attributes

- Nodes
  - Spatial coordinates
  - Spatial extents
  - Color
  - Shape
- Edges
  - Color
  - Shape
  - Width
  - Style



Visual Attributes

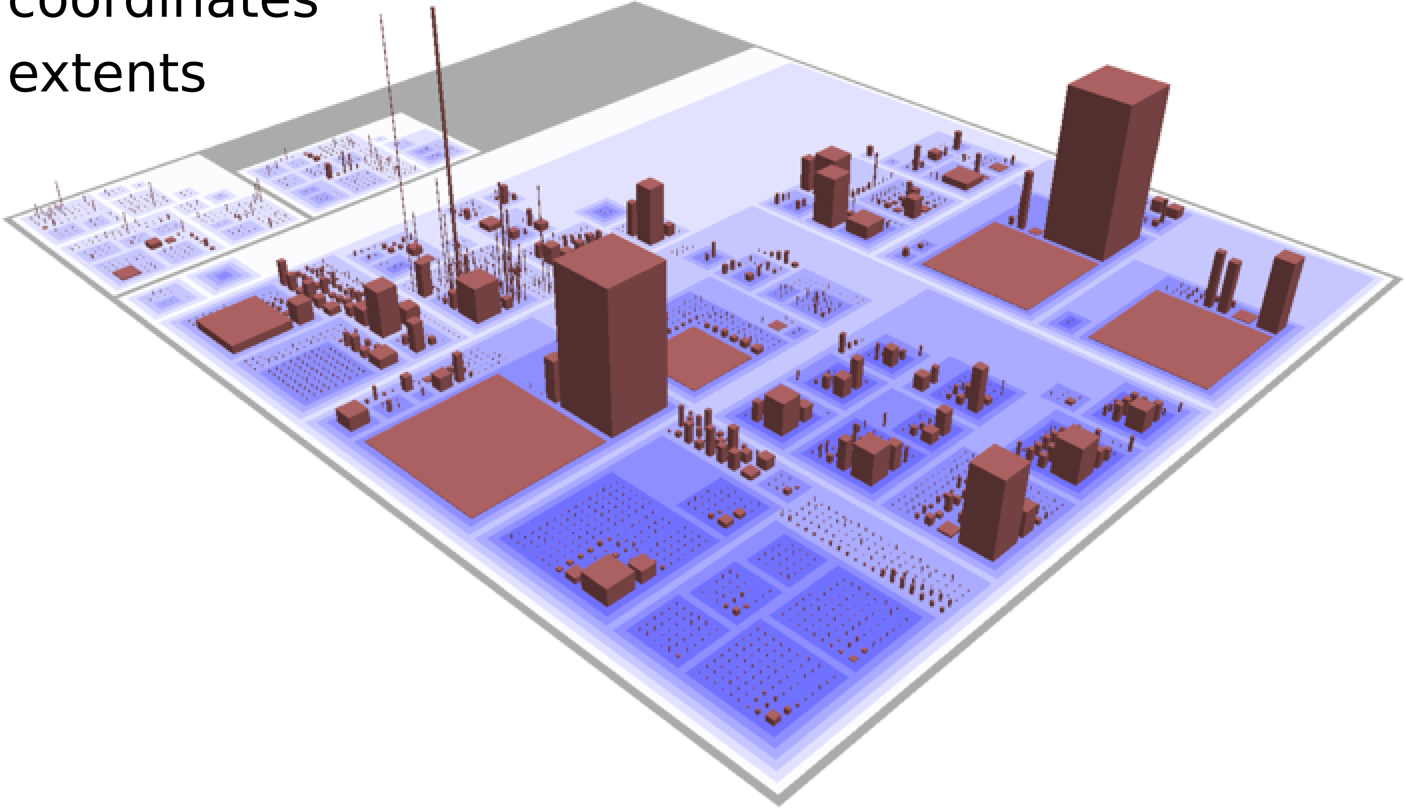
- Nodes
  - Spatial coordinates
  - Spatial extents
  - Color
  - Shape



- Edges
  - Color
  - Shape
  - Width
  - Style

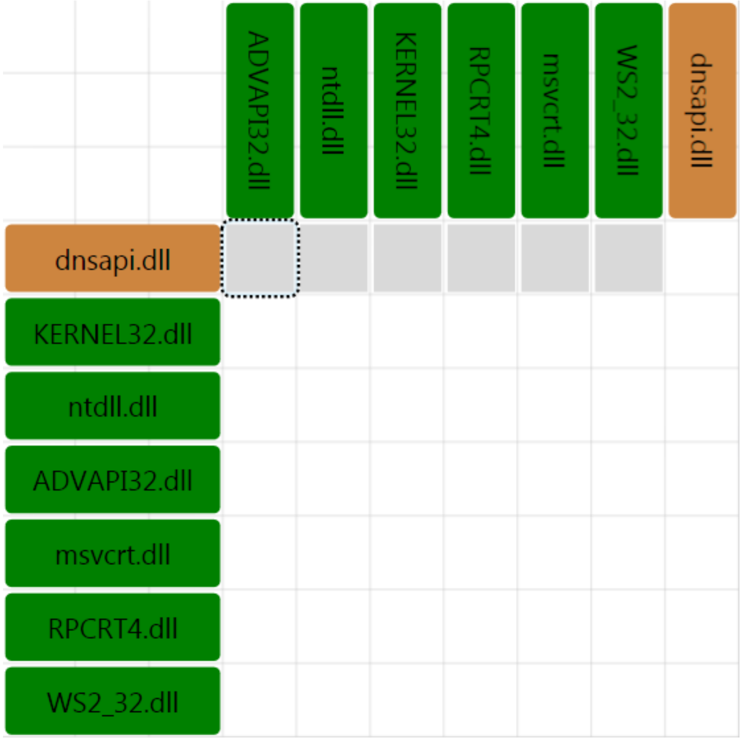
Visual Attributes

- Nodes
  - Spatial coordinates
  - Spatial extents
  - Color
  - Shape
- Edges
  - Color
  - Shape
  - Width
  - Style



Visualizing Software Security

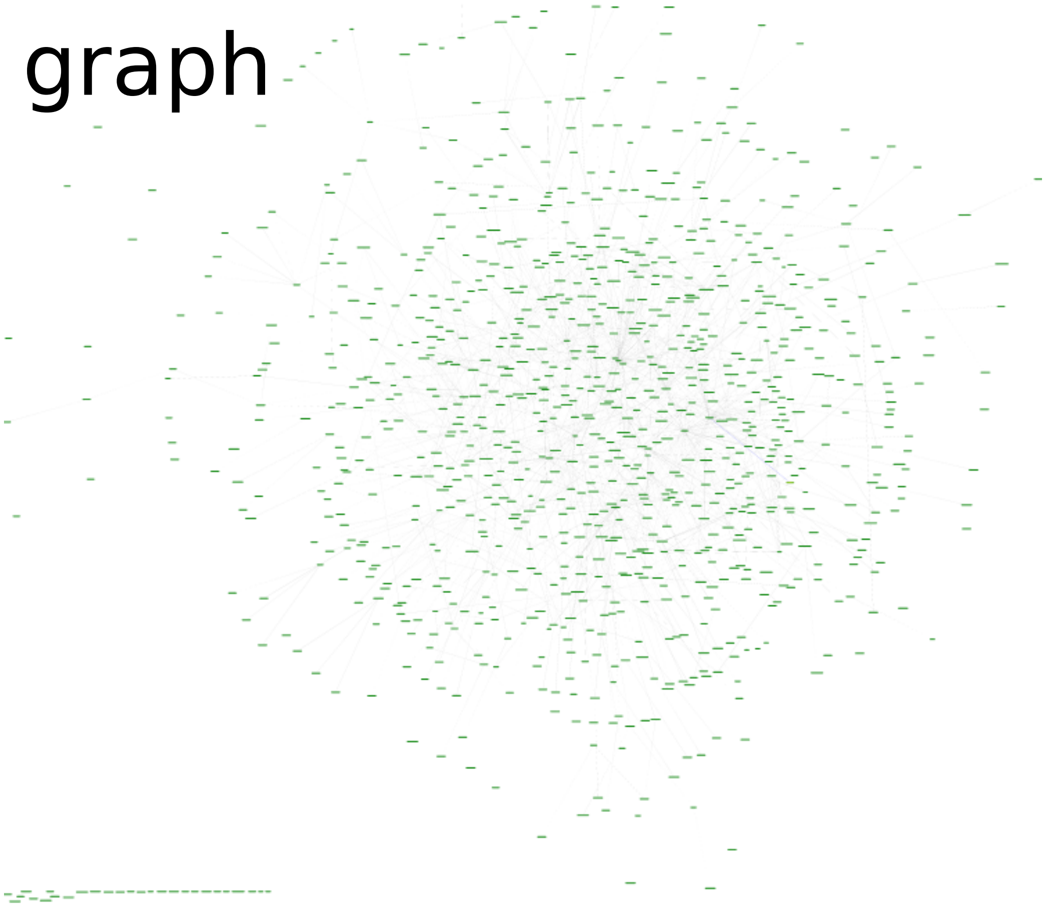
- Observe binary interdependencies



	ADVAPI32.dll	ntdll.dll	KERNEL32.dll	RPCRT4.dll	msvcrt.dll	WS2_32.dll	dnsapi.dll
dnsapi.dll	▨	▨	▨	▨	▨	▨	
KERNEL32.dll							
ntdll.dll							
ADVAPI32.dll							
msvcrt.dll							
RPCRT4.dll							
WS2_32.dll							

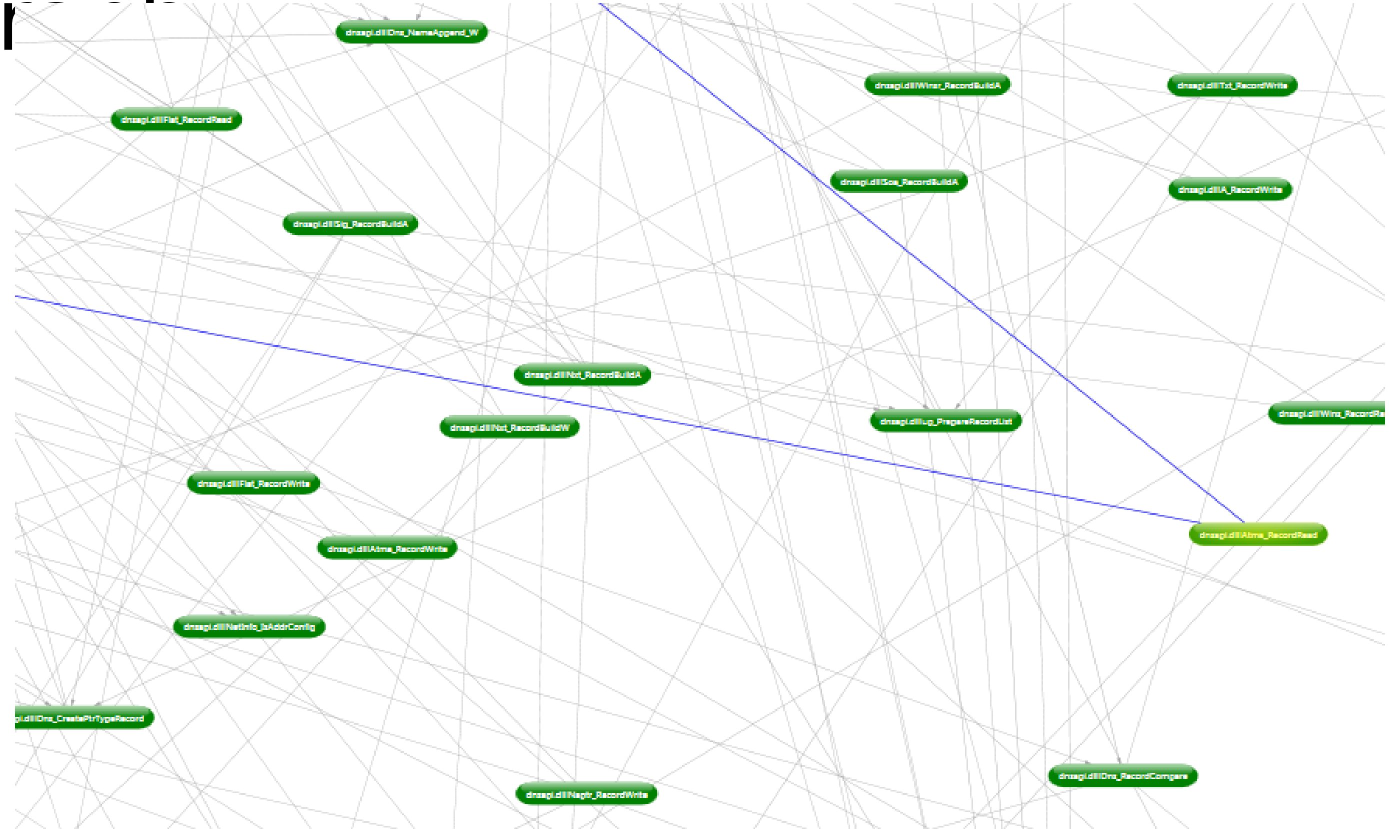
Visualizing Software Security

- Acquire a method level control flow graph



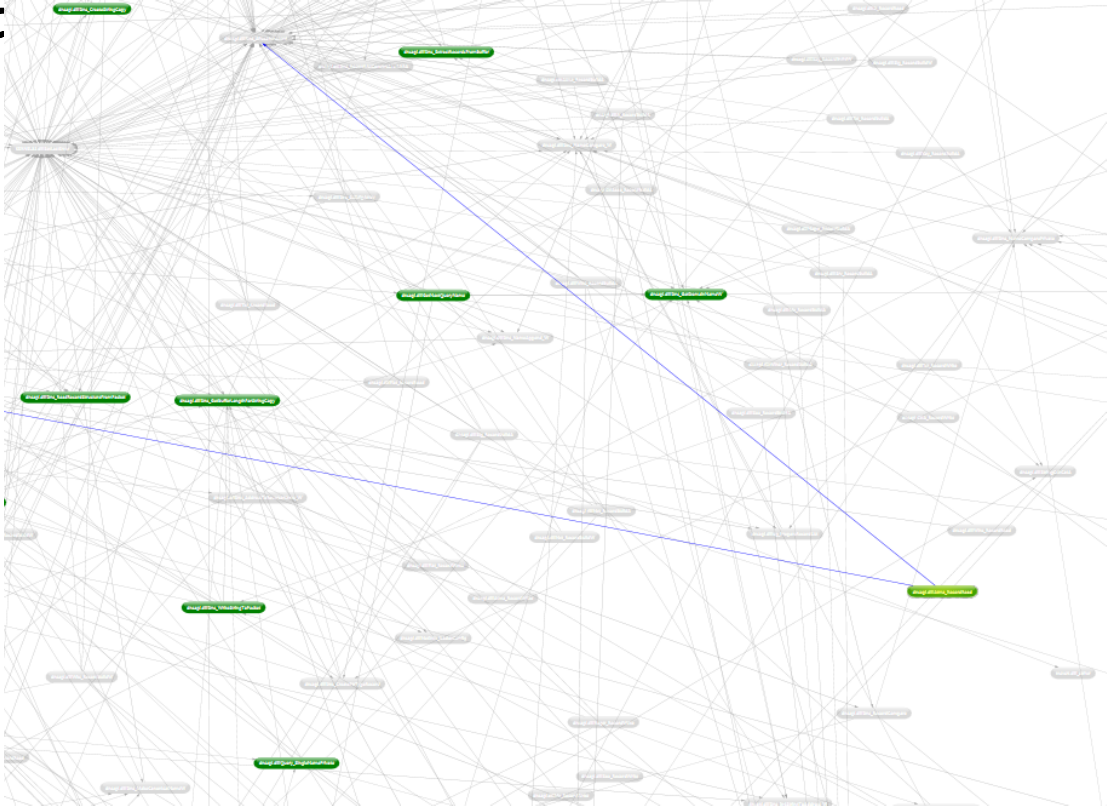
Visualizing Software Security

- Acquire a method level control flow gr


dnsapi.dll!Dns_NameAppend_W
dnsapi.dll!Winsr_RecordBuildA
dnsapi.dll!Txt_RecordWrite
dnsapi.dll!Flat_RecordRead
dnsapi.dll!Soa_RecordBuildA
dnsapi.dll!A_RecordWrite
dnsapi.dll!Sig_RecordBuildA
dnsapi.dll!Nxt_RecordBuildA
dnsapi.dll!up_PrepareRecordList
dnsapi.dll!Wins_RecordRe
dnsapi.dll!Nxt_RecordBuildW
dnsapi.dll!Flat_RecordWrite
dnsapi.dll!Atma_RecordRead
dnsapi.dll!Atma_RecordWrite
dnsapi.dll!NetInfo_IsAddrConfig
pi.dll!Dns_CreatePtrTypeRecord
dnsapi.dll!Dns_RecordCompare
dnsapi.dll!Naptr_RecordWrite

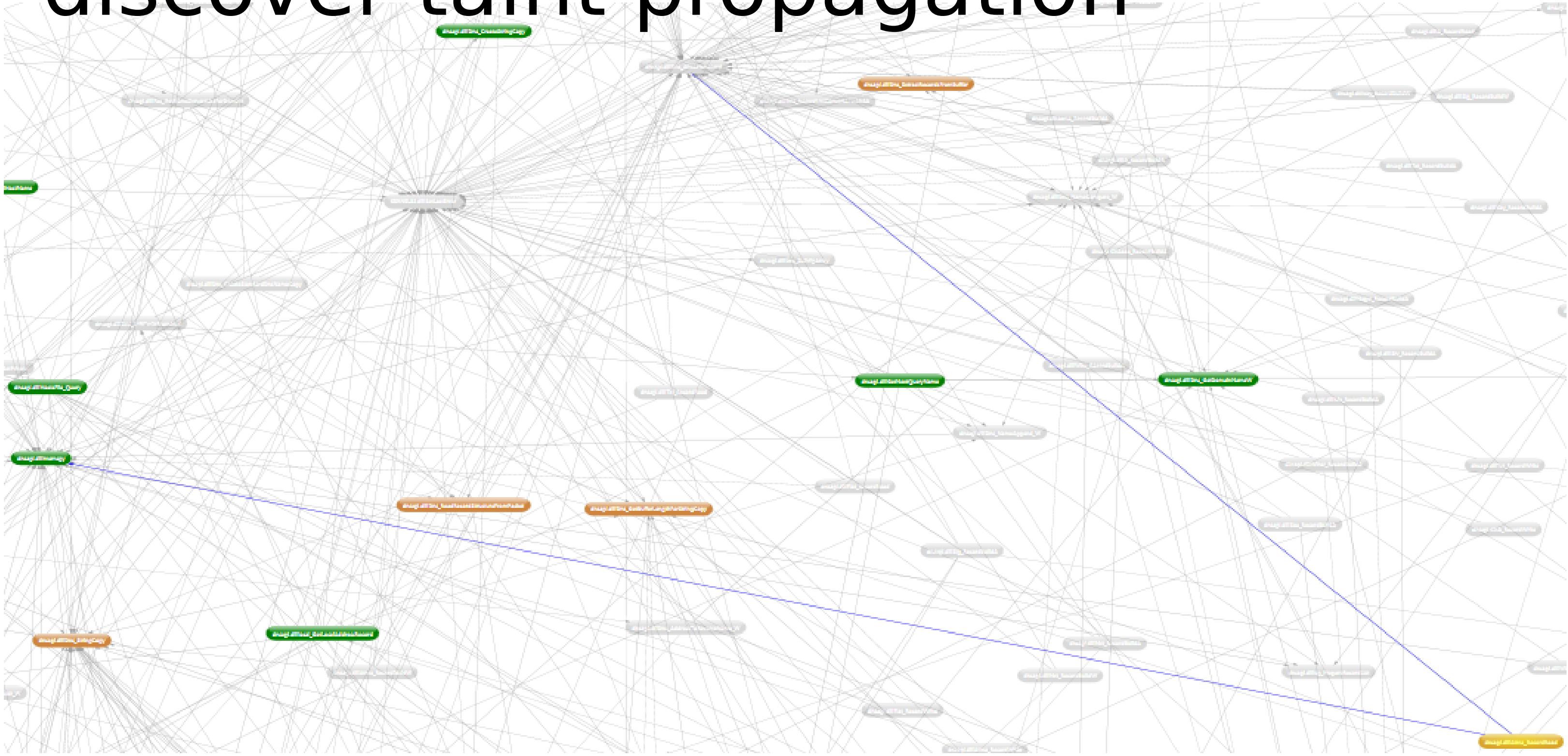
Visualizing Software Security

- Reduce graph using code coverage data



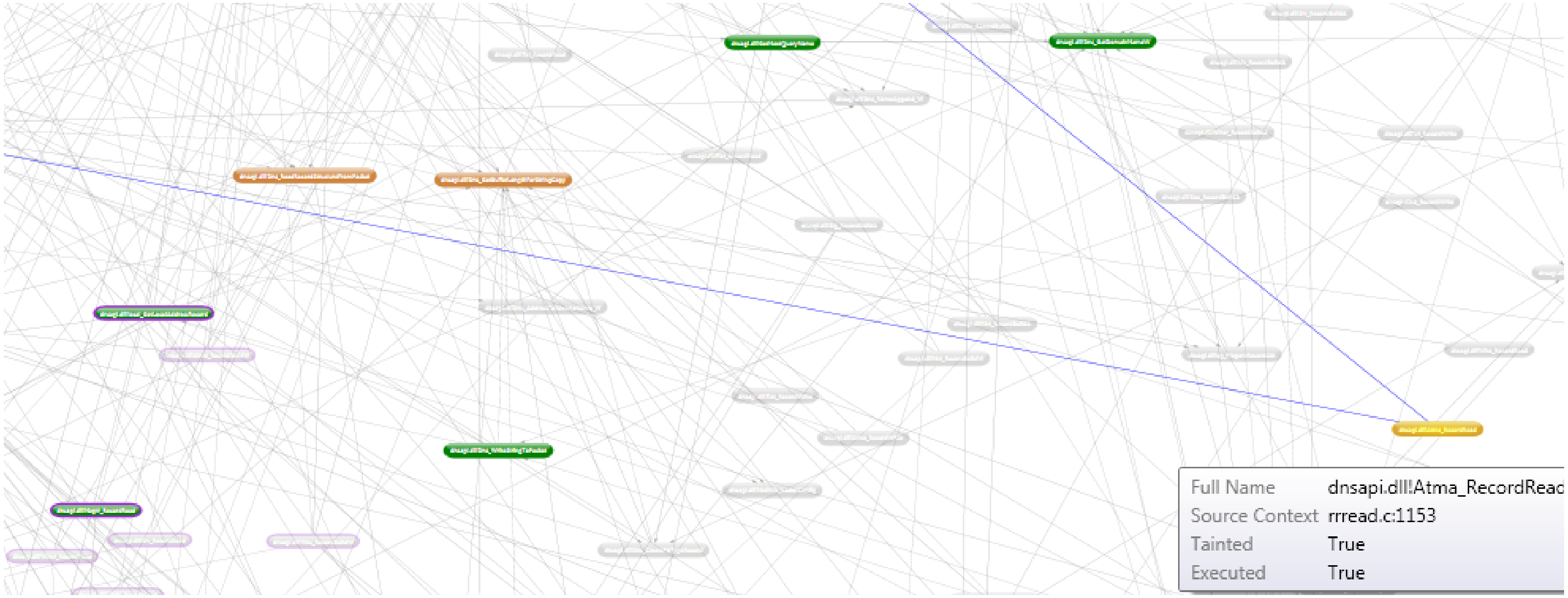
Visualizing Software Security

- Trace dataflow dependency to discover taint propagation



Visualizing Software Security

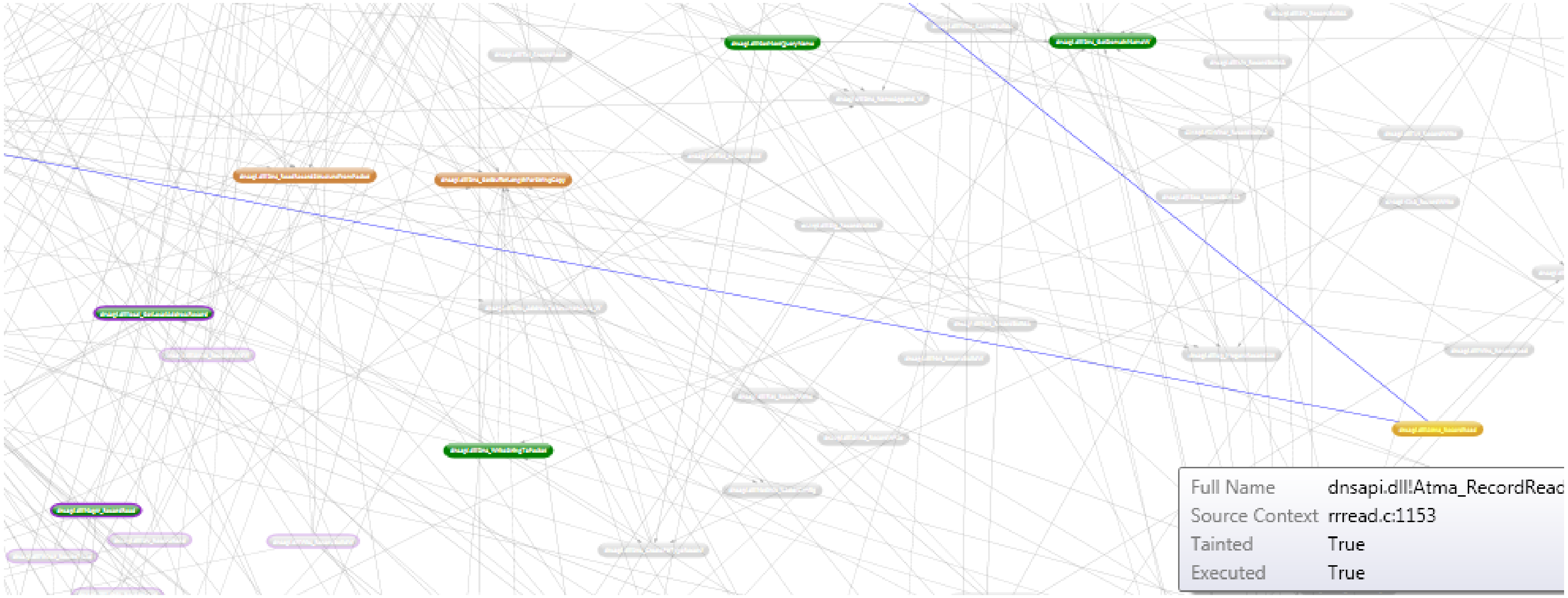
- Use static analysis plugins to derive security properties such as GS and SafeSEH



Full Name dnsapi.dll!Atma_RecordRead
Source Context rrread.c:1153
Tainted True
Executed True

Visualizing Software Security

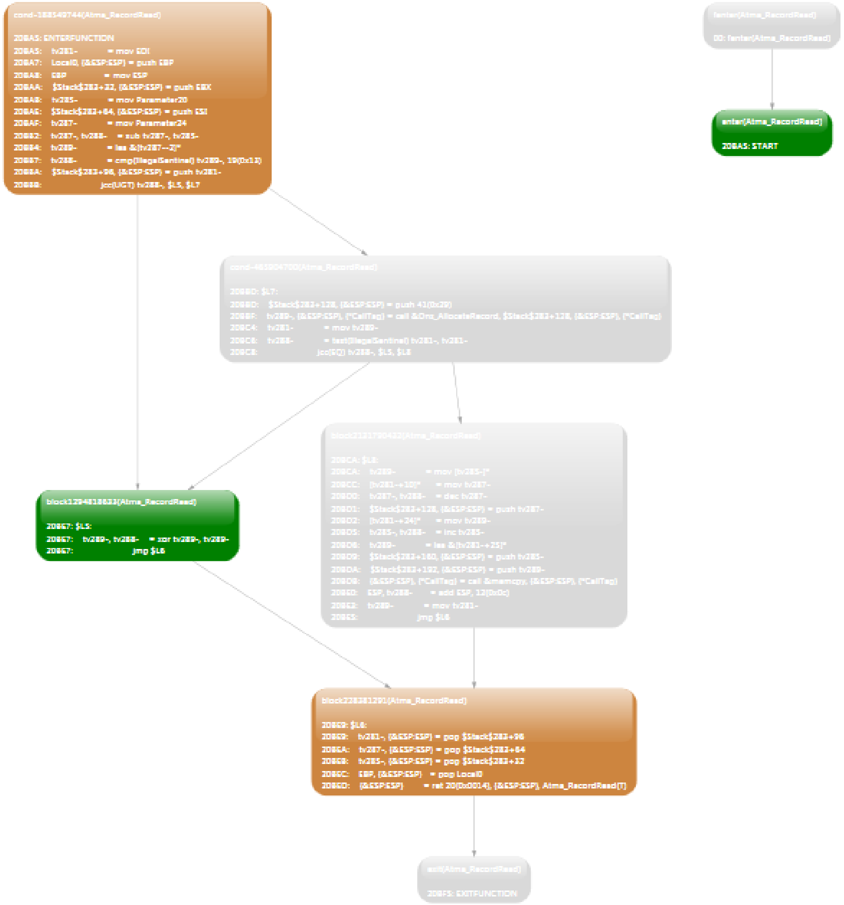
- Use static analysis plugins to derive security properties such as GS and SafeSEH



Full Name dnsapi.dll!Atma_RecordRead
Source Context rrread.c:1153
Tainted True
Executed True

Visualizing Software Security

- Analyze non-covered paths in tainted functions



```
fenter(Atma_RecordRead)

00: fenter(Atma_RecordRead)
```

```
enter(Atma_RecordRead)

208A5: START
```

```
cond-188549744(Atma_RecordRead)

208A5: ENTERFUNCTION
208A5:   tv281-              = mov EDI
208A7:   Local0, {&ESP:ESP} = push EBP
208A8:   EBP                 = mov ESP
208AA:   $Stack$282+32, {&ESP:ESP} = push EBX
208AB:   tv285-              = mov Parameter20
208AE:   $Stack$282+64, {&ESP:ESP} = push ESI
208AF:   tv287-              = mov Parameter24
208B2:   tv287-, tv288-     = sub tv287-, tv285-
208B4:   tv289-              = lea &[tv287-+-2]*
208B7:   tv288-              = cmp(IllegalSentinel) tv289-, 19(0x13)
208BA:   $Stack$282+96, {&ESP:ESP} = push tv281-
208BB:                       jcc(UGT) tv288-, $L5, $L7
```

```
cond-485804700(Atma_RecordRead)

208BD: $L7:
208BD:   $Stack$282+128, {&ESP:ESP} = push 41(0x29)
208BF:   tv289-, {&ESP:ESP}, {*CallTag} = call &Dns_AllocateRecord, $Stack$282+128, {&ESP:ESP}, {*CallTag}
208C4:   tv281-              = mov tv289-
208C6:   tv288-              = test(IllegalSentinel) tv281-, tv281-
208C8:                       jcc(EQ) tv288-, $L5, $L8
```

```
block1294818633(Atma_RecordRead)

208E7: $L5:
208E7:   tv289-, tv288-     = xor tv289-, tv289-
208E7:                       jmp $L6
```

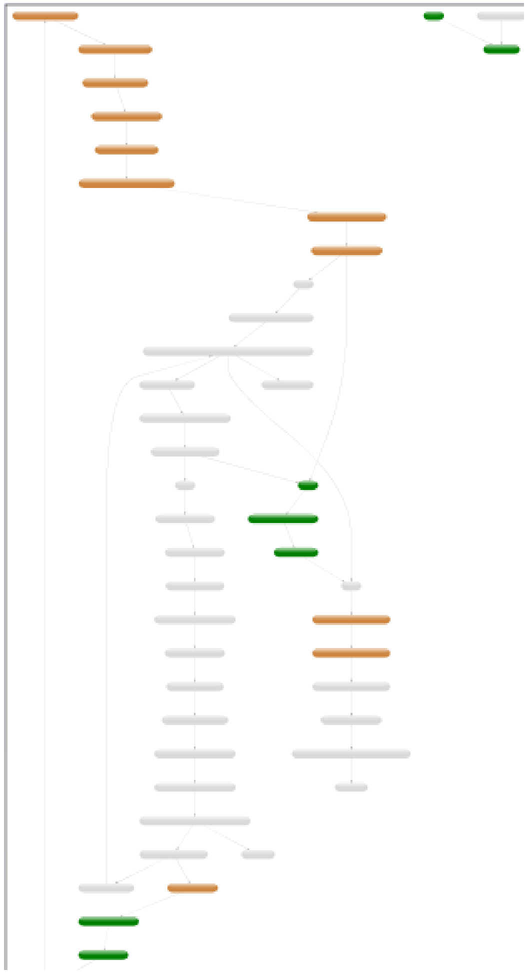
```
block2121790432(Atma_RecordRead)

208CA: $L8:
208CA:   tv289-              = mov [tv285-]*
208CC:   [tv281-+10]*        = mov tv287-
208D0:   tv287-, tv288-     = dec tv287-
208D1:   $Stack$282+128, {&ESP:ESP} = push tv287-
208D2:   [tv281-+24]*        = mov tv289-
208D5:   tv285-, tv288-     = inc tv285-
208D8:   tv289-              = lea &[tv281-+25]*
208D9:   $Stack$282+160, {&ESP:ESP} = push tv285-
208DA:   $Stack$282+192, {&ESP:ESP} = push tv289-
208DB:   {&ESP:ESP}, {*CallTag} = call &memcpy, {&ESP:ESP}, {*CallTag}
208E0:   ESP, tv288-        = add ESP, 12(0x0c)
208E3:   tv289-              = mov tv281-
208E5:                       jmp $L6
```

```
block228381291(Atma_RecordRead)

208E9: $L6:
208E9:   tv281-, {&ESP:ESP} = pop $Stack$282+96
208EA:   tv287-, {&ESP:ESP} = pop $Stack$282+64
208EB:   tv285-, {&ESP:ESP} = pop $Stack$282+32
208EC:   EBP, {&ESP:ESP}    = pop Local0
208ED:   {&ESP:ESP}          = ret 20(0x0014), {&ESP:ESP}, Atma_RecordRead(T)
```

```
exit(Atma_RecordRead)

208F3: EXITFUNCTION
```

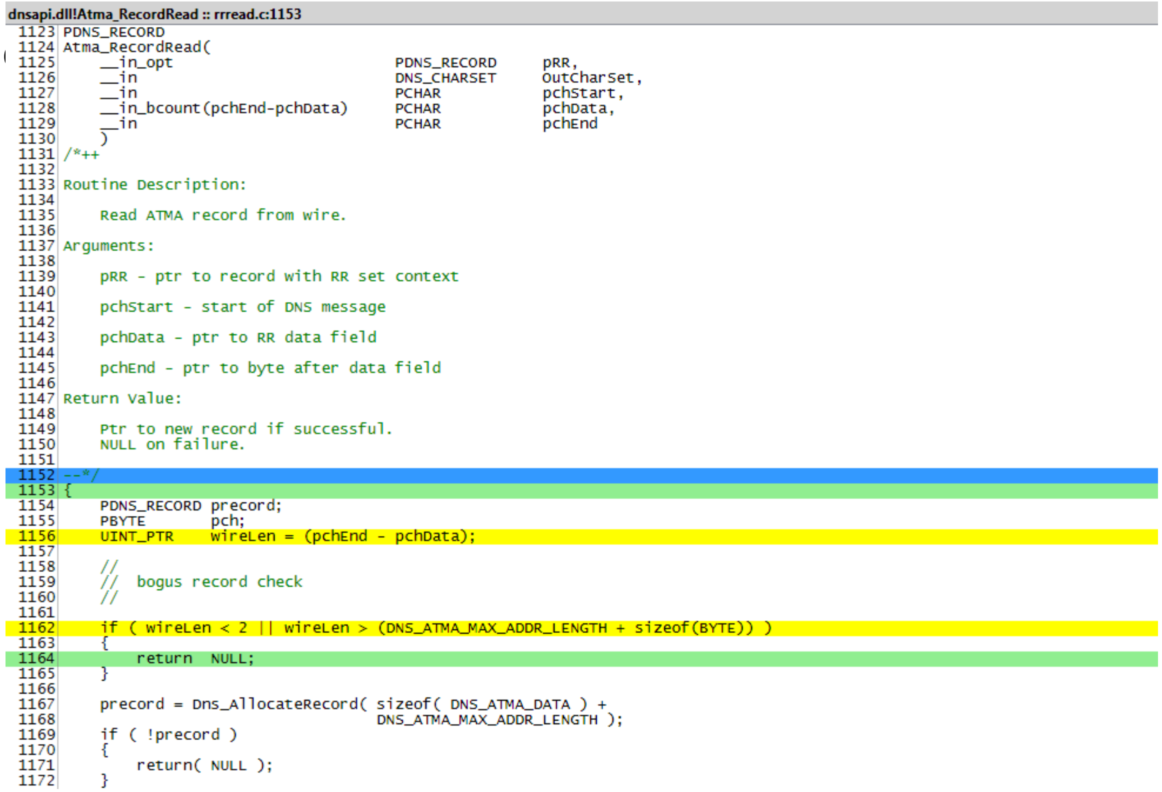
Visualizing Software Security

- Analyze non-covered paths in tainted functions



Visualizing Software Properties

- Examine source code where correl
  occur

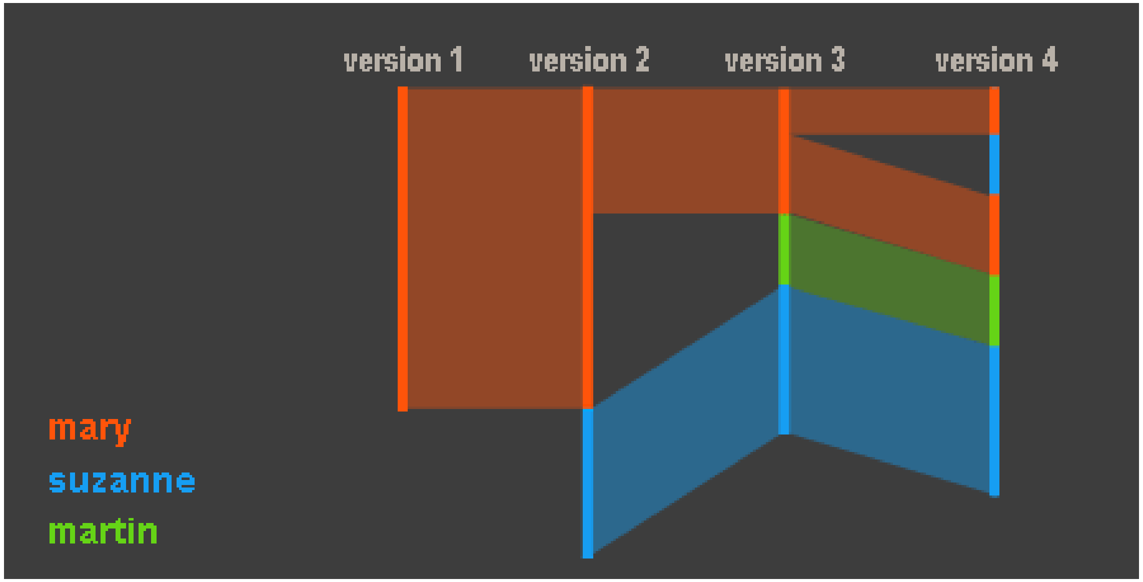


dnsapi.dll!Atma_RecordRead :: rrread.c:1153

```
PDNS_RECORD
Atma_RecordRead(
    __in_opt                        PDNS_RECORD     pRR,
    __in                            DNS_CHARSET     OutCharSet,
    __in                            PCHAR           pchStart,
    __in_bcount(pchEnd-pchData)     PCHAR           pchData,
    __in                            PCHAR           pchEnd
    )
/*++

Routine Description:

    Read ATMA record from wire.

Arguments:

    pRR - ptr to record with RR set context

    pchStart - start of DNS message

    pchData - ptr to RR data field

    pchEnd - ptr to byte after data field

Return Value:

    Ptr to new record if successful.
    NULL on failure.

--*/
{
    PDNS_RECORD precord;
    PBYTE       pch;
    UINT_PTR    wireLen = (pchEnd - pchData);

    //
    //  bogus record check
    //

    if ( wireLen < 2 || wireLen > (DNS_ATMA_MAX_ADDR_LENGTH + sizeof(BYTE)) )
    {
        return  NULL;
    }

    precord = Dns_AllocateRecord( sizeof( DNS_ATMA_DATA ) +
                                  DNS_ATMA_MAX_ADDR_LENGTH );
    if ( !precord )
    {
        return( NULL );
    }
```

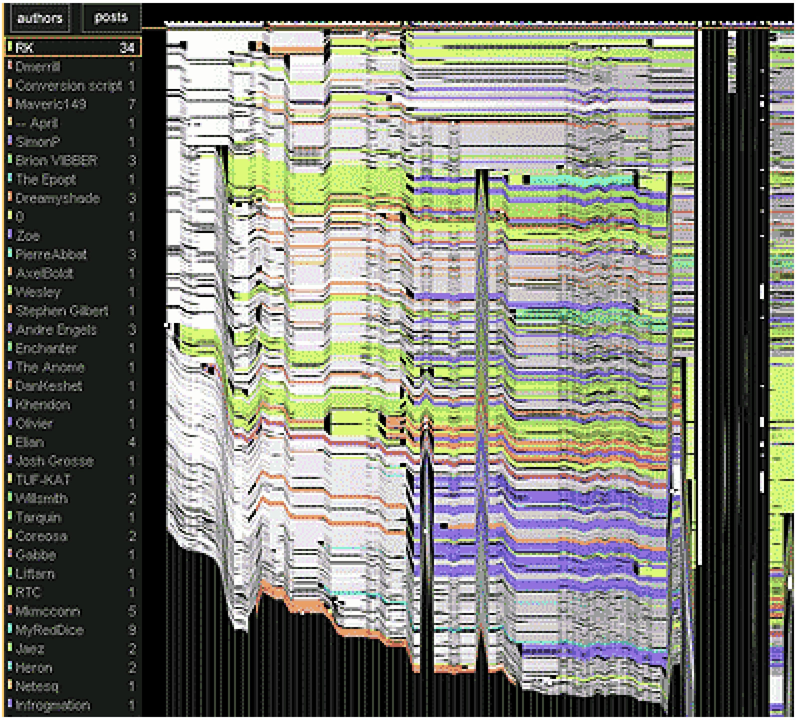
Beyond Graphs

- Source Code Revision History
  - History Flow

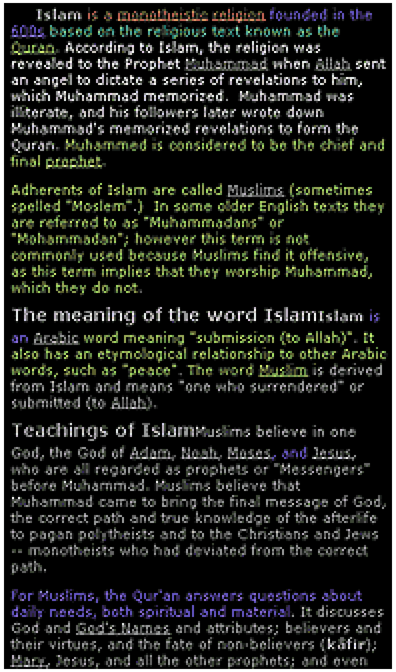


Beyond Graphs

- Source Code Revision History
  - History Flow



authors	posts
RK	34
Dmerrill	1
Conversion script	1
Maveric149	7
-- April	1
SimonP	1
Brion VIBBER	3
The Epopt	1
Dreamyshade	3
0	1
Zoe	1
PierreAbbat	3
AxelBoldt	1
Wesley	1
Stephen Gilbert	1
Andre Engels	3
Enchanter	1
The Anome	1
DanKeshet	1
Khendon	1
Olivier	1
Ellan	4
Josh Grosse	1
TUF-KAT	1
Willsmith	2
Tarquin	1
Coreosa	2
Gabbe	1
Liftarn	1
RTC	1
Mkmcconn	5
MyRedDice	9
Jaez	2
Heron	2
Netesq	1
Infrogmation	1



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. Muhammed 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.

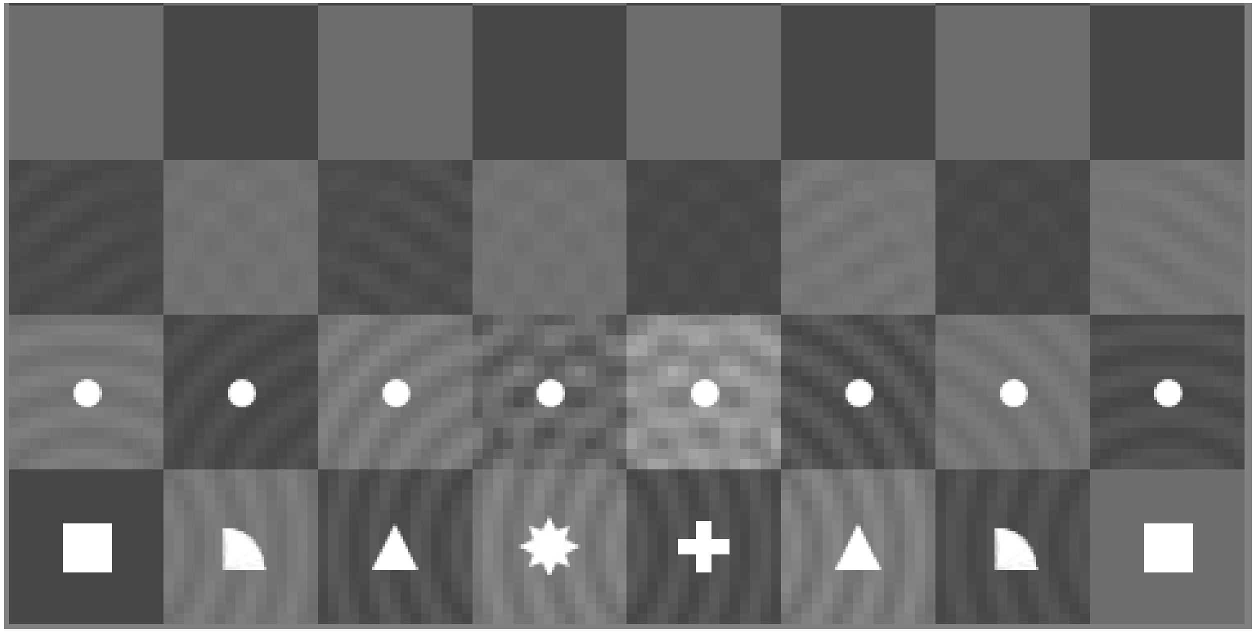
The meaning of the word IslamIslam is an Arabic word meaning "submission (to Allah)". It also has an etymological relationship to other Arabic words, such as "peace". The word Muslim is derived from Islam and means "one who surrendered" or submitted (to Allah).

Teachings of IslamMuslims 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 (kâfir); Mary, Jesus, and all the other prophets; and even

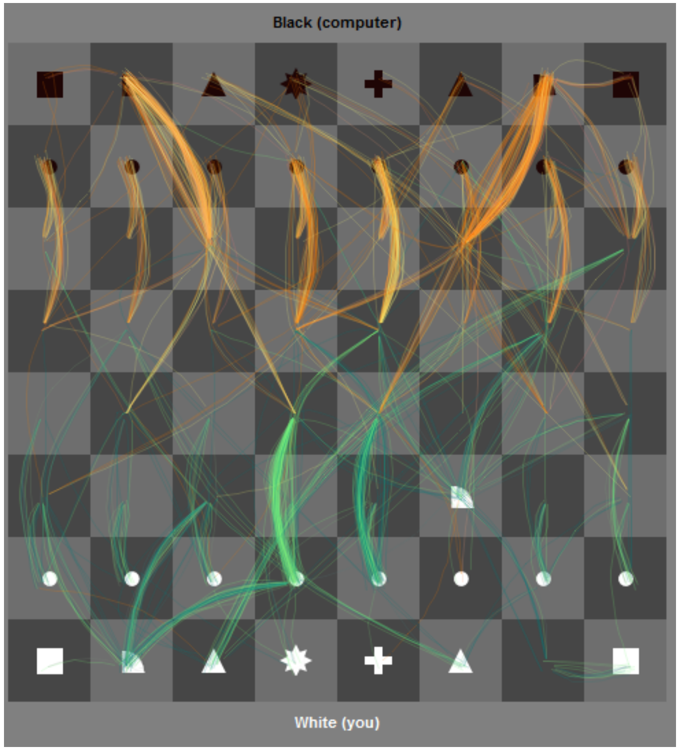
Beyond Graphs

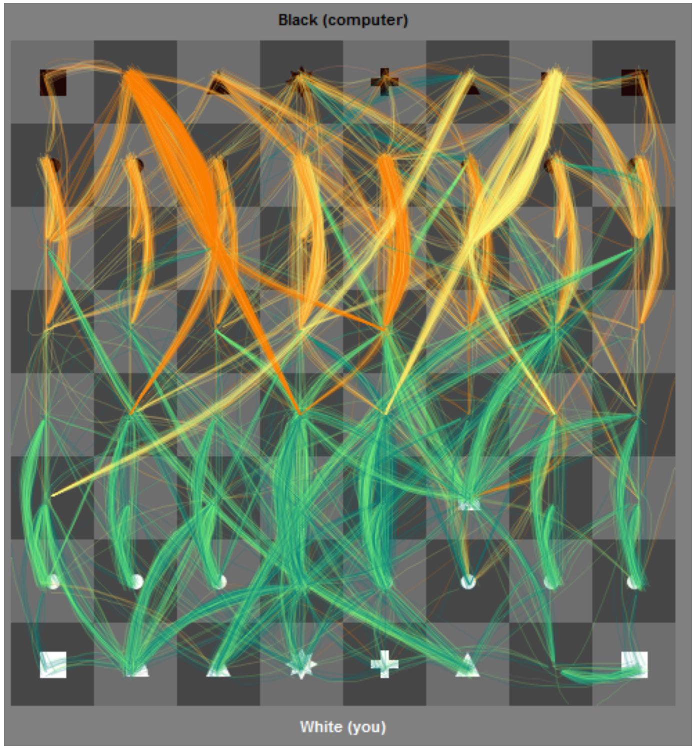
- State Machine Models
  - Thinking Machine



Beyond Graphs

- State Machine Models
  - Thinking Machine





Questions?

Richard Johnson
richardj@microsoft.com

Thank you!

http://swiscience

alias: pandora

Richard Johnson
richardj@microsoft.com