Common Weakness Enumeration (CWE)

1.9 Status update

Conor Harris
Software Systems Engineer, Senior
MITRE Corporation

Making Security Measurable™

Acknowledgements

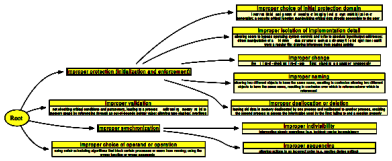
- Various MSM Contributors
  - Robert Martin, CWE Project Lead
  - Steve Christey, CWE Technical Lead
  - Sean Barnum, CAPEC Lead

Common Weakness Enumeration (CWE)

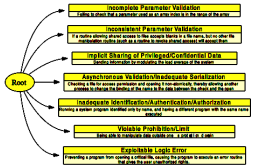
- **dictionary of weaknesses**
  - **weaknesses that can lead to exploitable vulnerabilities (i.e. CVEs)**
  - **the things we don't want in our code, design, or architecture**
  - **web site with XML of content, sources of content, and process used**
- **structured views**
  - **currently provide hierarchical view into CWE dictionary content**
  - **will evolve to support alternate views**
- **open community process**
  - **to facilitate common terms/ concepts/facts and understanding**
  - **allows for vendors, developers, system owners and acquirers to understand tool capabilities/ coverage and priorities**
  - **utilize community expertise**

Foundation for other DHS, NSA, OSD, NIST, OWASP, SANS, and OMG SwA Efforts

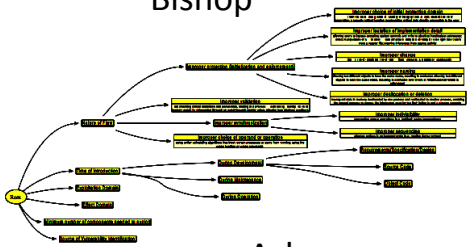
Protection Analysis



RISOS

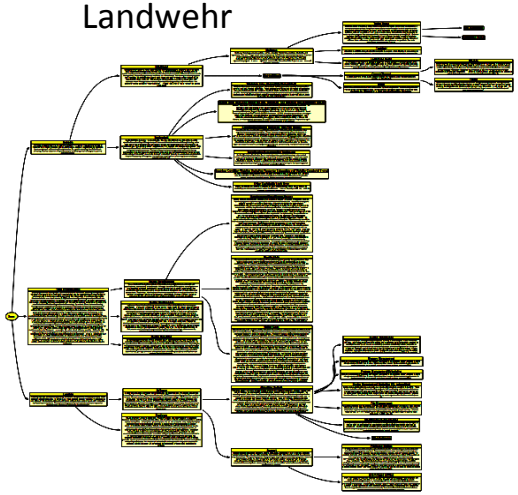


Bishop

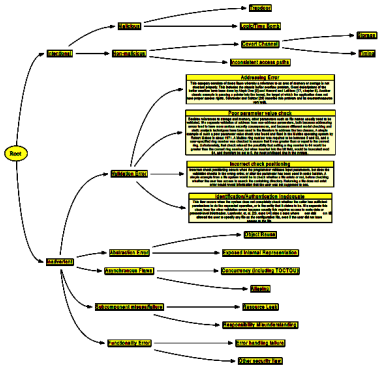


Aslam

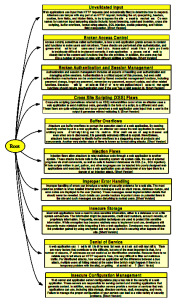
Landwehr



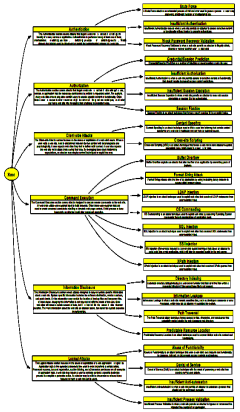
Weber



OWASP



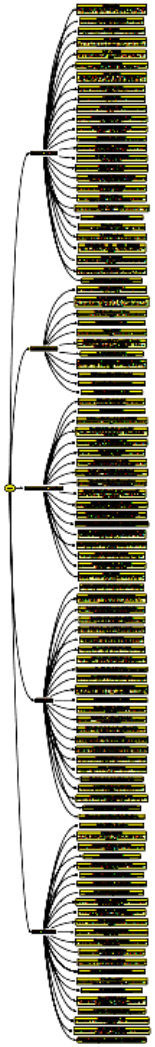
WASC



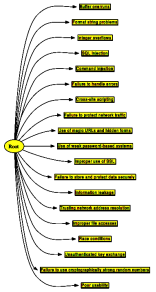
Tool A

Tool B

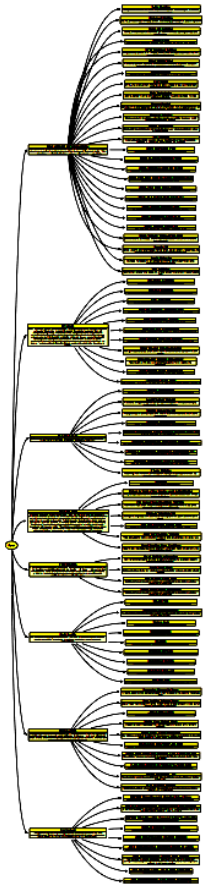
CLASP



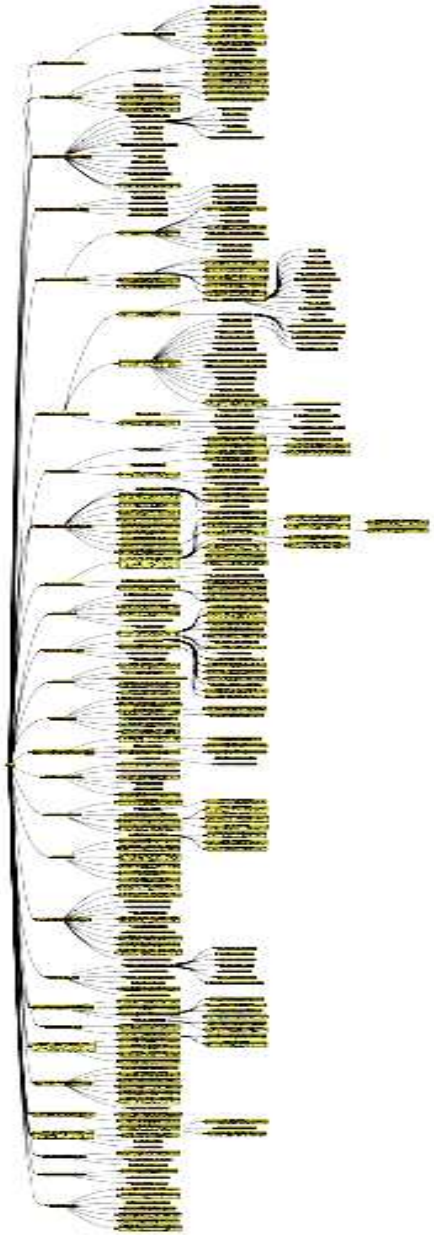
Microsoft



7 Kingdoms

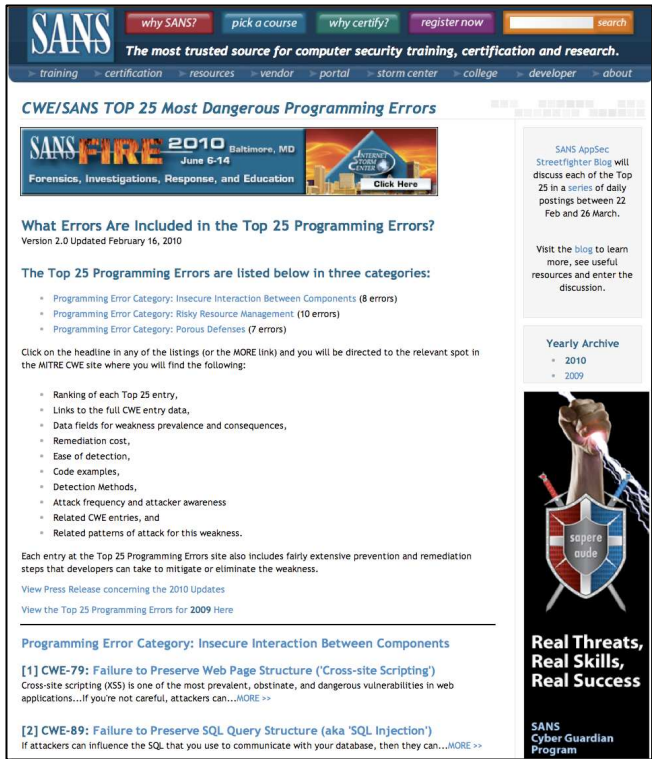


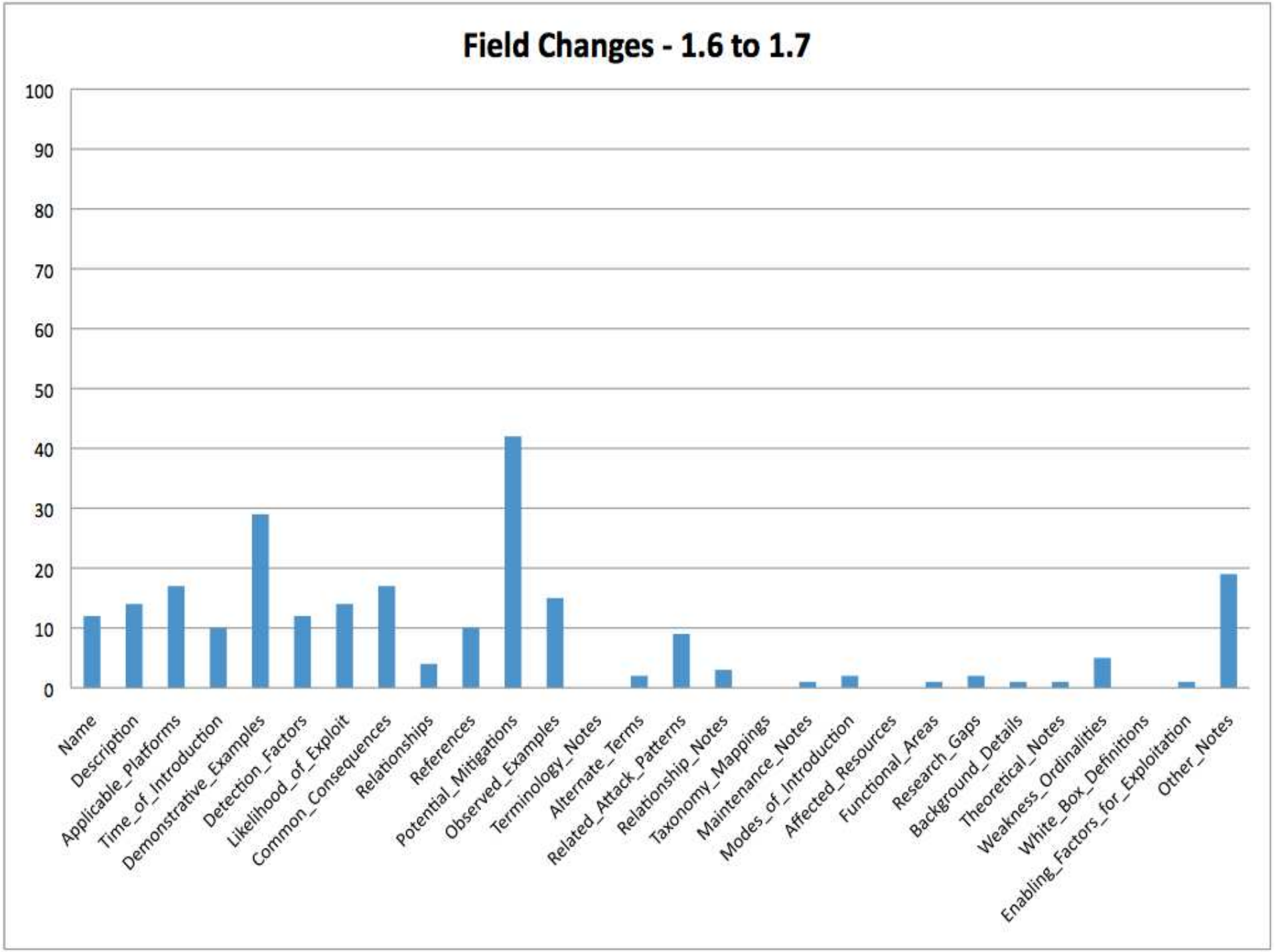
PLOVER

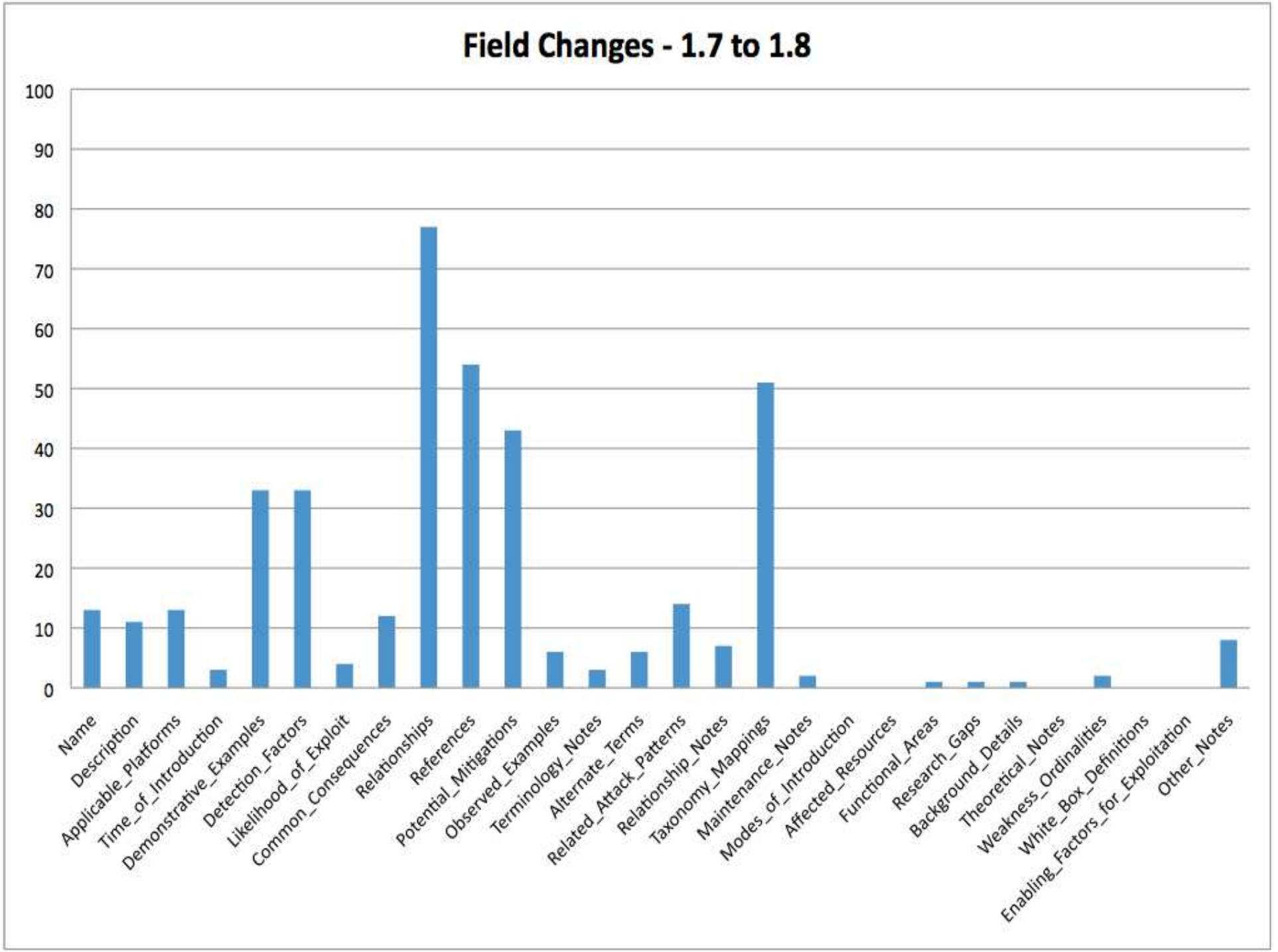


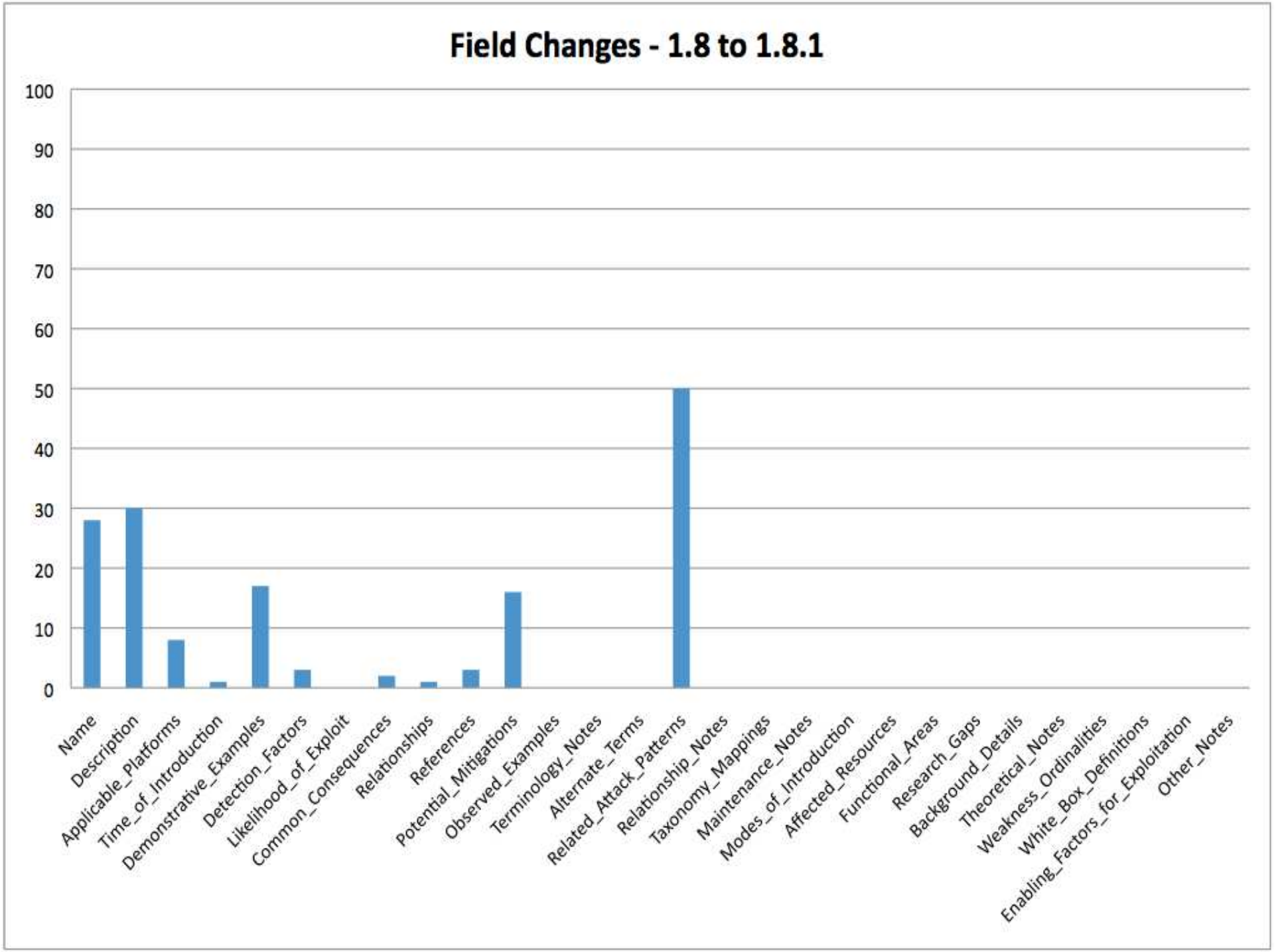
20010 CWE/SANS Top 25 Programming Errors

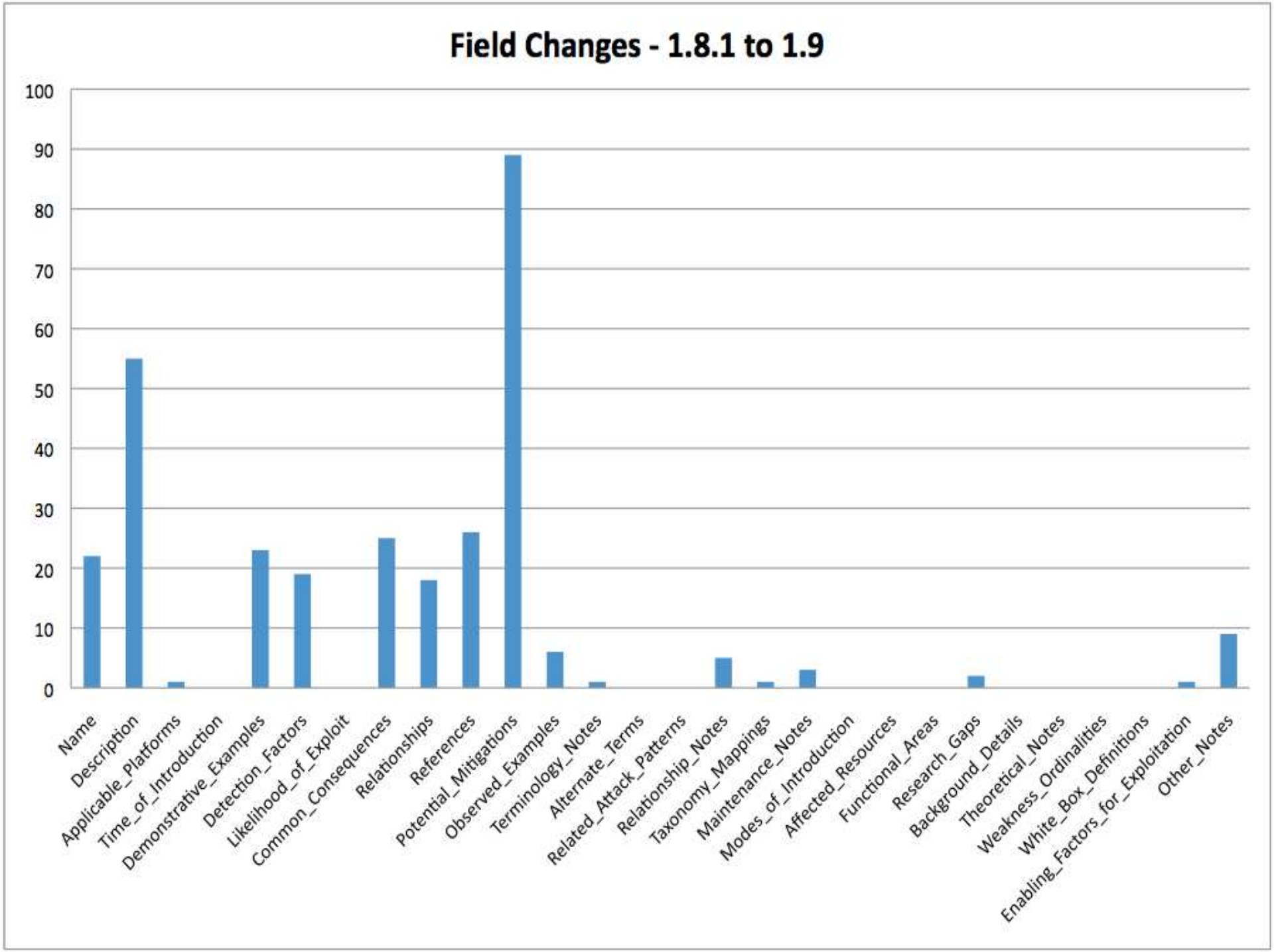
- **Sponsored by:**
  - **National Cyber Security Division (DHS)**
- **List was selected by a group of security experts from 34 organizations including:**
  - **Academia: Purdue, Northern Kentucky University**
  - **Government: CERT, NSA, DHS**
  - **Software Vendors: Microsoft, Oracle, Red Hat, Apple, Juniper, McAfee, Symantec, Sun, RSA (of EMC)**
  - **Security Vendors: Veracode, Fortify, Cigital, Mandiant, Cigital, SRI, Secunia, Breach, SAIC, Aspect, WhiteHat**
  - **Security Groups: OWASP, WASC**




Field Changes - 1.6 to 1.7
100
90
80
70
60
50
40
30
20
10
0
Name
Description
Applicable_Platforms
Time_of_Introduction
Demonstrative_Examples
Detection_Factors
Likelihood_of_Exploit
Common_Consequences
Relationships
References
Potential_Mitigations
Observed_Examples
Terminology_Notes
Alternate_Terms
Related_Attack_Patterns
Relationship_Notes
Taxonomy_Mappings
Maintenance_Notes
Modes_of_Introduction
Affected_Resources
Functional_Areas
Research_Gaps
Background_Details
Theoretical_Notes
Weakness_Ordinalities
White_Box_Definitions
Enabling_Factors_for_Exploitation
Other_Notes


Field Changes - 1.7 to 1.8
100
90
80
70
60
50
40
30
20
10
0
Name
Description
Applicable_Platforms
Time_of_Introduction
Demonstrative_Examples
Detection_Factors
Likelihood_of_Exploit
Common_Consequences
Relationships
References
Potential_Mitigations
Observed_Examples
Terminology_Notes
Alternate_Terms
Related_Attack_Patterns
Relationship_Notes
Taxonomy_Mappings
Maintenance_Notes
Modes_of_Introduction
Affected_Resources
Functional_Areas
Research_Gaps
Background_Details
Theoretical_Notes
Weakness_Ordinalities
White_Box_Definitions
Enabling_Factors_for_Exploitation
Other_Notes


Field Changes - 1.8 to 1.8.1
100
90
80
70
60
50
40
30
20
10
0
Name
Description
Applicable_Platforms
Time_of_Introduction
Demonstrative_Examples
Detection_Factors
Likelihood_of_Exploit
Common_Consequences
Relationships
References
Potential_Mitigations
Observed_Examples
Terminology_Notes
Alternate_Terms
Related_Attack_Patterns
Relationship_Notes
Taxonomy_Mappings
Maintenance_Notes
Modes_of_Introduction
Affected_Resources
Functional_Areas
Research_Gaps
Background_Details
Theoretical_Notes
Weakness_Ordinalities
White_Box_Definitions
Enabling_Factors_for_Exploitation
Other_Notes


Field Changes - 1.8.1 to 1.9
100
90
80
70
60
50
40
30
20
10
0
Name
Description
Applicable_Platforms
Time_of_Introduction
Demonstrative_Examples
Detection_Factors
Likelihood_of_Exploit
Common_Consequences
Relationships
References
Potential_Mitigations
Observed_Examples
Terminology_Notes
Alternate_Terms
Related_Attack_Patterns
Relationship_Notes
Taxonomy_Mappings
Maintenance_Notes
Modes_of_Introduction
Affected_Resources
Functional_Areas
Research_Gaps
Background_Details
Theoretical_Notes
Weakness_Ordinalities
White_Box_Definitions
Enabling_Factors_for_Exploitation
Other_Notes

Mitigation Library

- Normalize common mitigations across Top 25
  - E.g. general “Input Validation” mitigation can be applied across various injection issues (79, 89, 78, 129...) can be mitigated w/ validation
  - Analyze all Top 25 mits, merge overlap
  - Why is this good?
    - Higher quality
    - Easier to fill in (new entries and gaps)
    - Single point of maintenance

Example: MIT-4.1

GENERAL

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

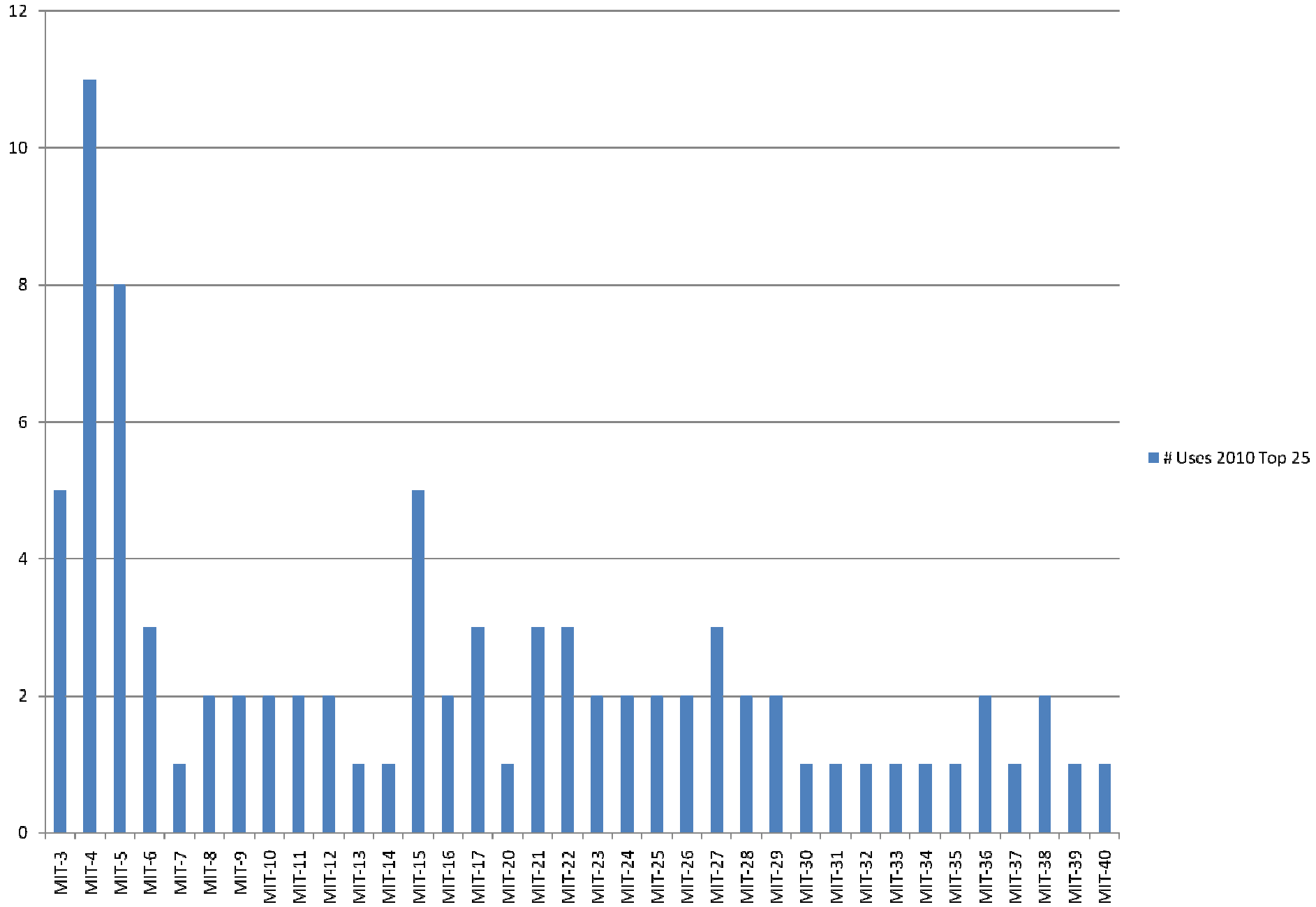
SPECIFIC

Examples include the Safe C String Library (SafeStr) by Messier and Viega, and the Strsafe.h library from Microsoft. These libraries provide safer versions of overflow-prone string-handling functions. This is not
a complete solution, since many buffer overflows are not related to strings.

Heavy focus on Top 25

- Uses 36 of the 41 mitigations in the library

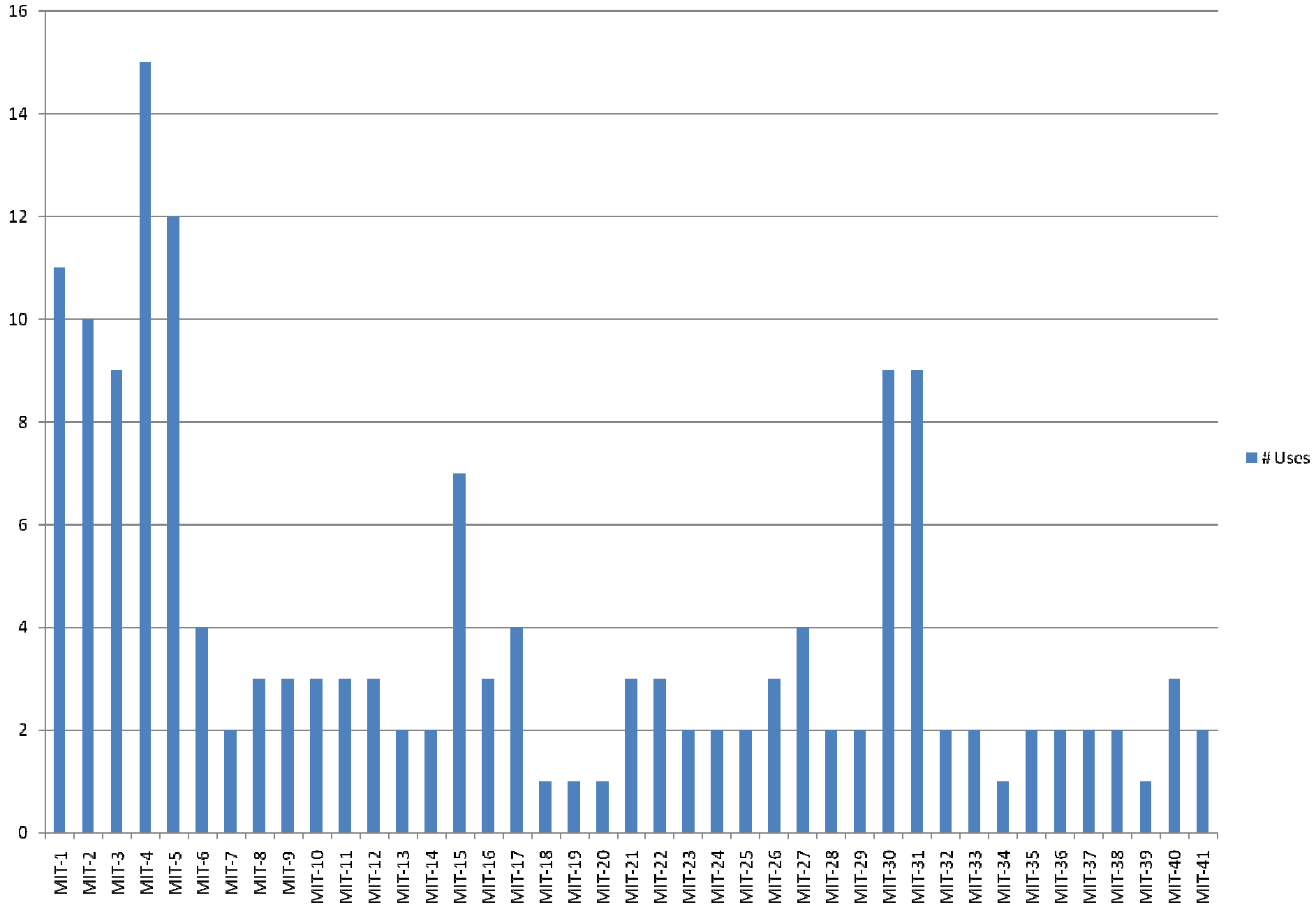
- 85 occurrences, for an average of 3.4 Top 25 entries per mitigation


Uses 2010 Top 25
12
10
8
6
4
2
0
MIT-3
MIT-4
MIT-5
MIT-6
MIT-7
MIT-8
MIT-9
MIT-10
MIT-11
MIT-12
MIT-13
MIT-14
MIT-15
MIT-16
MIT-17
MIT-20
MIT-21
MIT-22
MIT-23
MIT-24
MIT-25
MIT-26
MIT-27
MIT-28
MIT-29
MIT-30
MIT-31
MIT-32
MIT-33
MIT-34
MIT-35
MIT-36
MIT-37
MIT-38
MIT-39
MIT-40
Uses 2010 Top 25

Most frequently used across CWE merged

- 159 total library mitigations (41 unique)

- Less than 60% of usage is in Top 25
  - Spread easily
  - “Second” use case
    - XSS


Mitlib Uses All CWE
16
14
12
10
8
6
4
2
0
MIT-1
MIT-2
MIT-3
MIT-4
MIT-5
MIT-6
MIT-7
MIT-8
MIT-9
MIT-10
MIT-11
MIT-12
MIT-13
MIT-14
MIT-15
MIT-16
MIT-17
MIT-18
MIT-19
MIT-20
MIT-21
MIT-22
MIT-23
MIT-24
MIT-25
MIT-26
MIT-27
MIT-28
MIT-29
MIT-30
MIT-31
MIT-32
MIT-33
MIT-34
MIT-35
MIT-36
MIT-37
MIT-38
MIT-39
MIT-40
MIT-41
Uses

Not Perfect…

- Copy + paste burden still exists
  - Backup checks in place
- What if only the impacted resource varies?
  - e.g.
    - use a reference map for safe URLs
    - use a reference map for safe file names
- Work in progress

More Mitigation Work

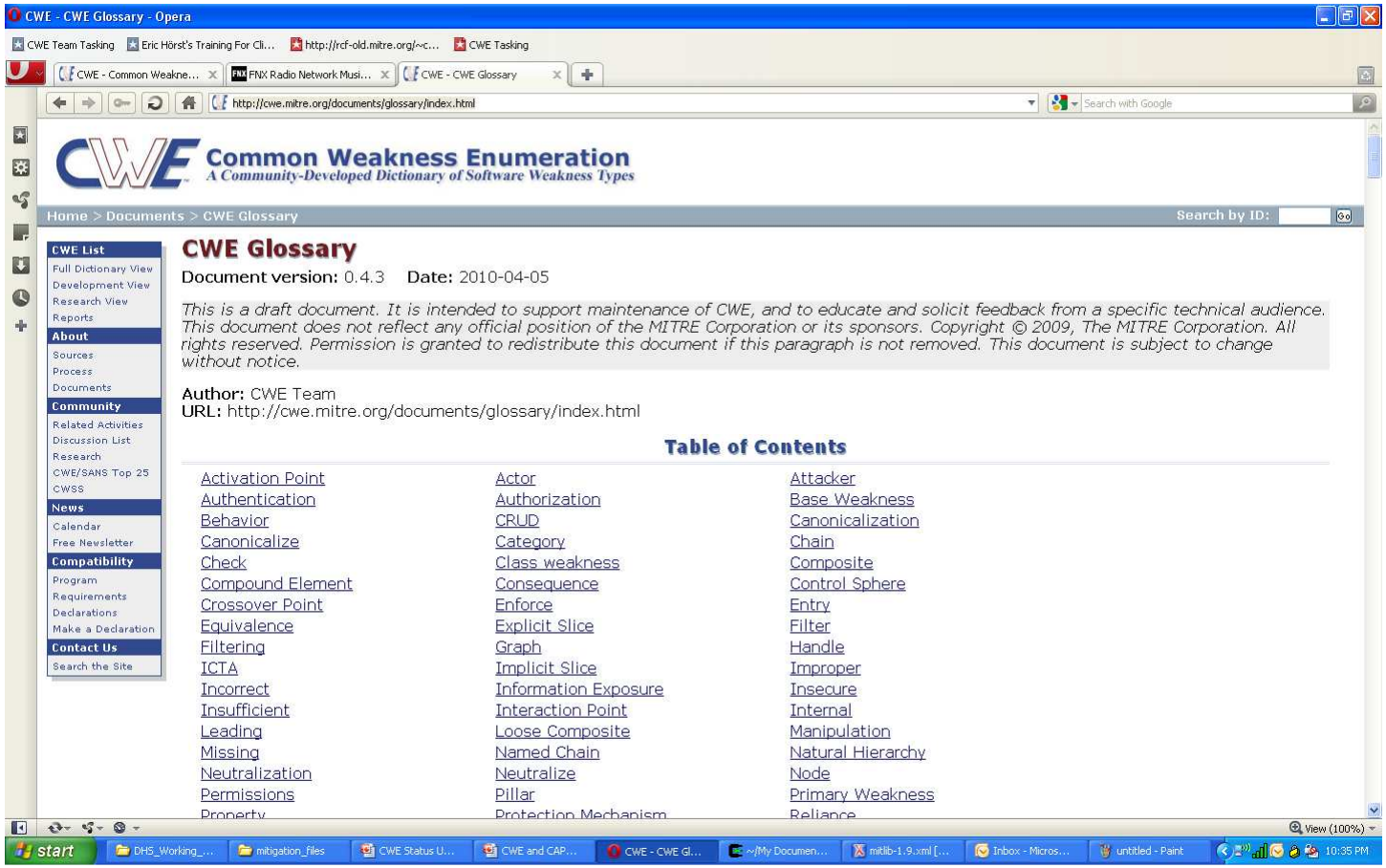
- Top 25 Mitigation Strategies
  - Enumeration
    - e.g. Input Validation, Language Selection, Identify & Reduce Attack Surface
  - Machine detectable

Top 25 Consequences

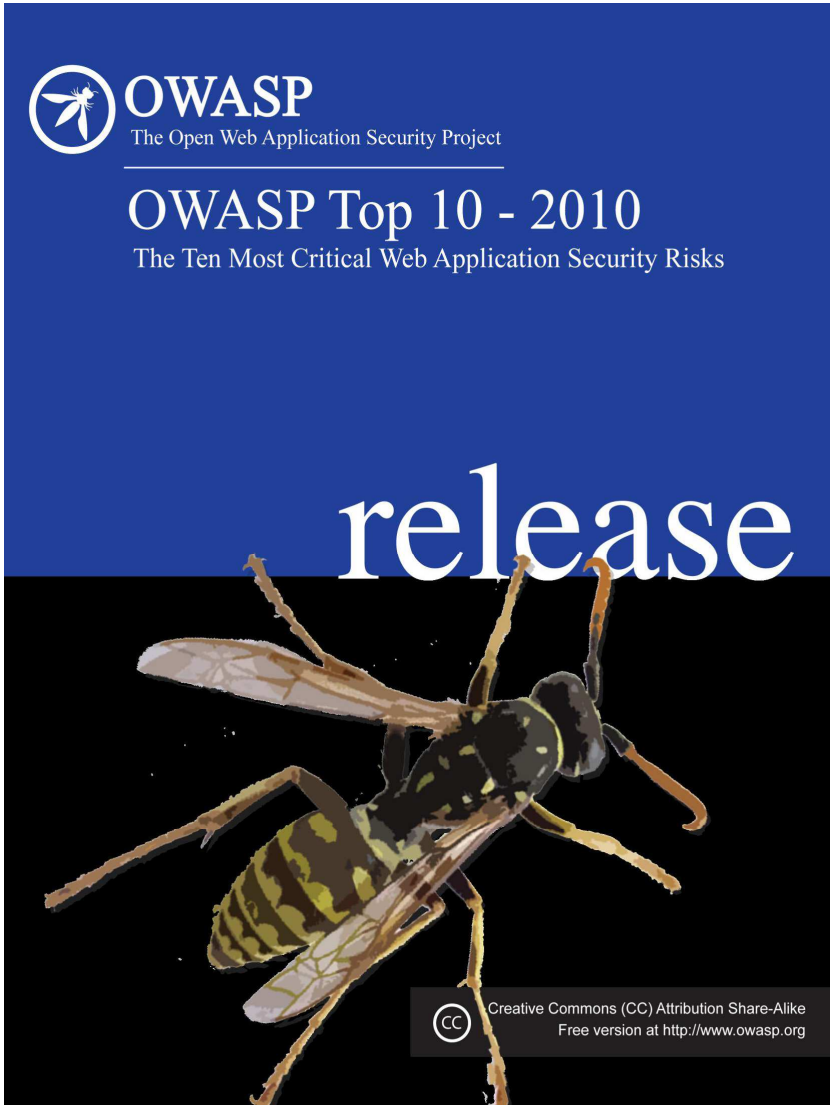
- Some standard content
- Technical Impacts
  - More specific than CIA
  - Allows for further automatic mapping
    - Feeds into pocket guide work
  - How to describe technical impact of information exposure?
    - Array indexing issue -> Read/Write mem
    - What about nabbing cookies via XSS?
    - Getting pathname via error message info exposure?

Vocabulary Work

- http://cwe.mitre.org/documents/glossary/index.html
- Reduce usage of overloaded words



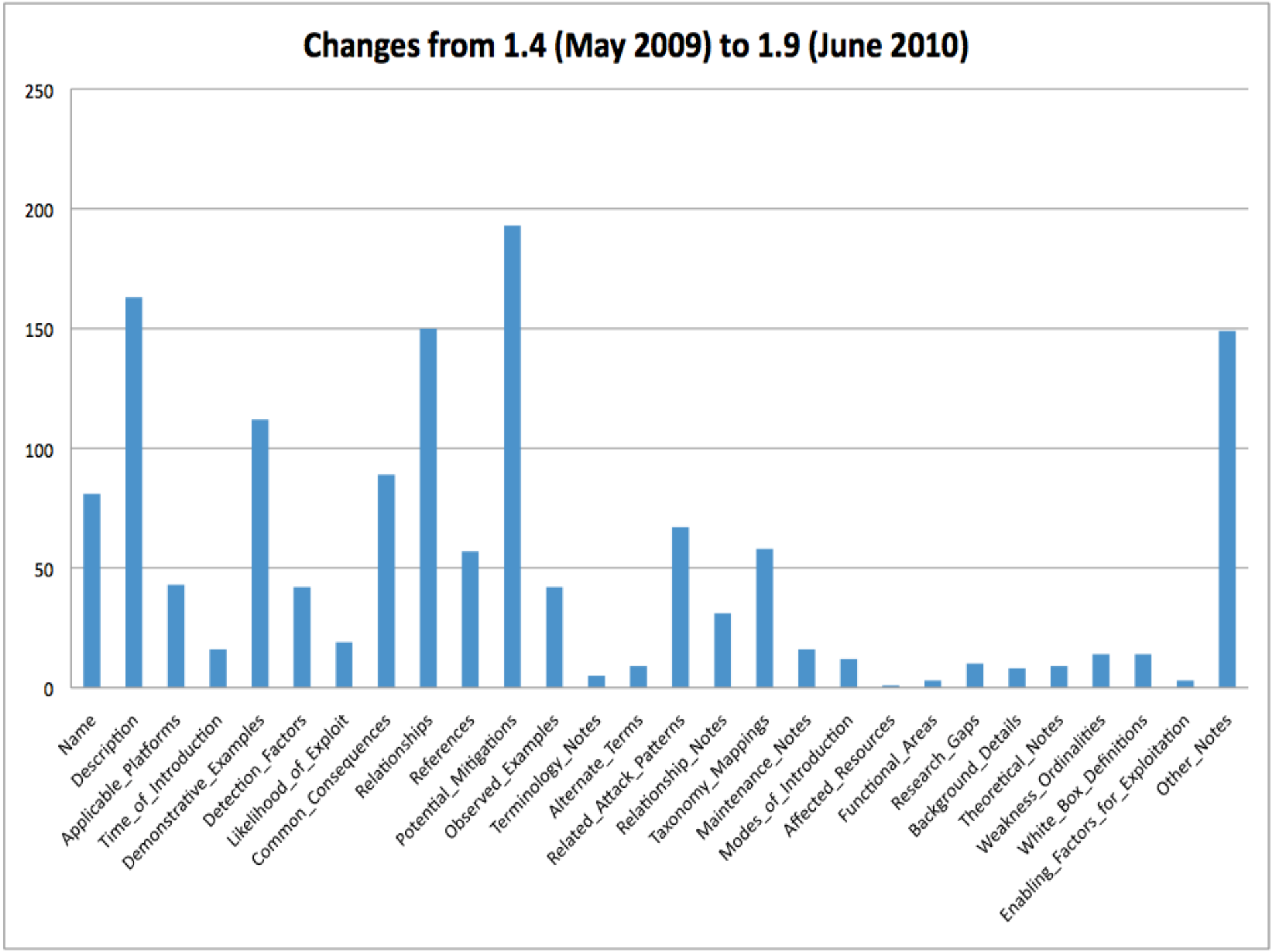
CWE-809 OWASP Top 10 2010

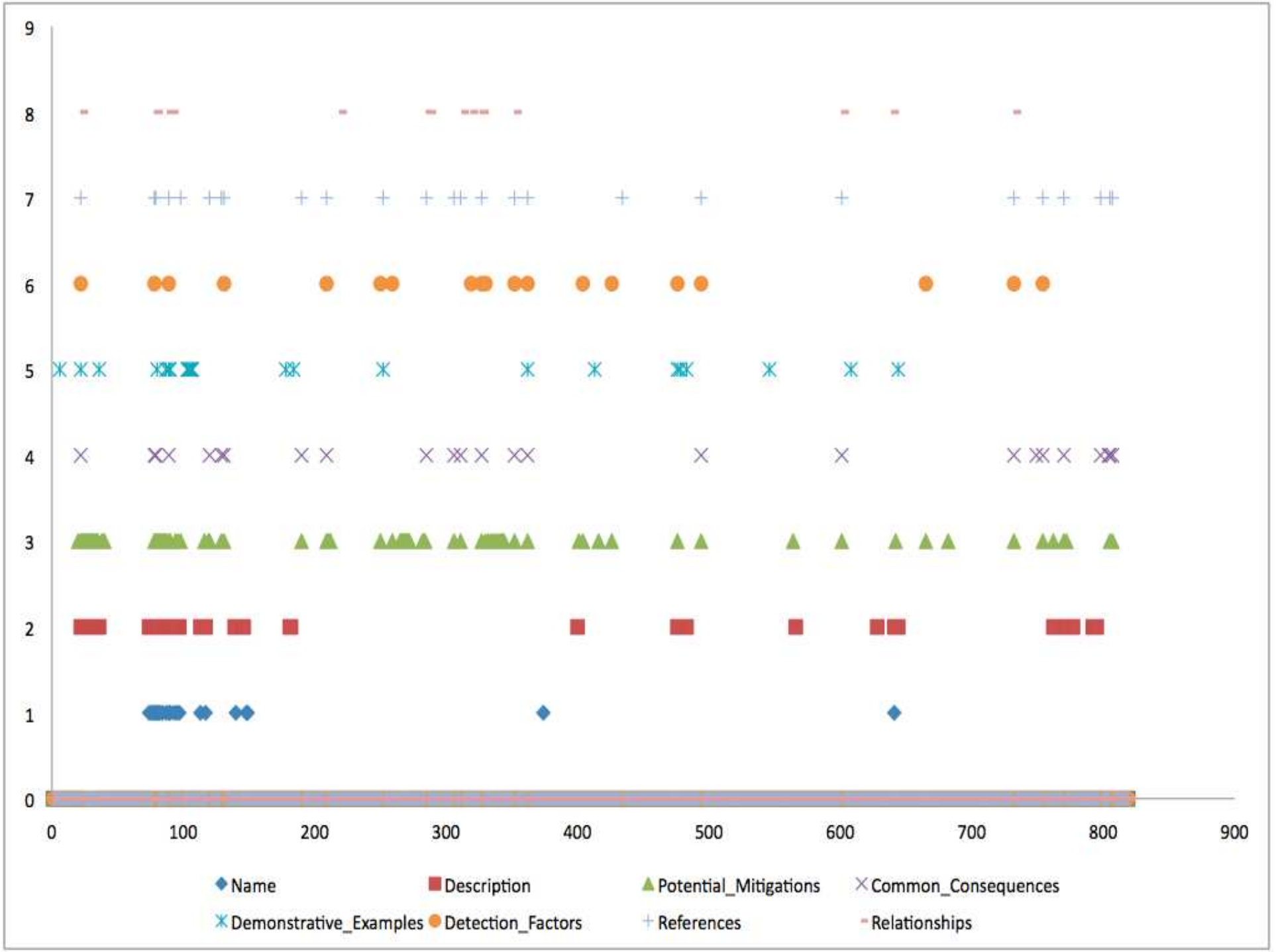


SANS Street Fighter Blog

Questions?

- Comments, criticism, questions welcome
  cwe@mitre.org


Changes from 1.4 (May 2009) to 1.9 (June 2010)
250
200
150
100
50
0
Name
Description
Applicable_Platforms
Time_of_Introduction
Demonstrative_Examples
Detection_Factors
Likelihood_of_Exploit
Common_Consequences
Relationships
References
Potential_Mitigations
Observed_Examples
Terminology_Notes
Alternate_Terms
Related_Attack_Patterns
Relationship_Notes
Taxonomy_Mappings
Maintenance_Notes
Modes_of_Introduction
Affected_Resources
Functional_Areas
Research_Gaps
Background_Details
Theoretical_Notes
Weakness_Ordinalities
White_Box_Definitions
Enabling_Factors_for_Exploitation
Other_Notes


9
8
7
6
5
4
3
2
1
0
0
100
200
300
400
500
600
700
800
900
Name
Description
Potential_Mitigations
Common_Consequences
Demonstrative_Examples
Detection_Factors
References
Relationships