Security Data Mining in an Ontology for Vulnerability Management

CVE used to describe vulnerabilities could be expressed with a simple Extended BNF (EBNF) as follows:

CVE\_VULNERABILITY ::=

( IT\_SYSTEM VERSION+ )+

|“because of ” REASON\*

|“when” CONDITION\*

|“in” (COMPONENT|FILE|FUNCTION)\*

“has” (specified|unspecified) VULNERABILITY

“allow” (specified|unspecified) ATTACKER+

“to cause” CONSEQUENCE+

|“via” ATTACK\*

Thus a vulnerability could be expressed as:

CVE-2008-5070.DESC ::=

Pro Chat Rooms 3.0.3

when magic\_quotes\_gpc is diabled

has the SQL Injection Vulnerability

allow remote attackers

to cause arbitrary SQL commands execution

via the guid parameter to (1)index.php and (2)admin.php

The rule of the **similarVulnerability** in SWRL:

Vulnerability(?x) ^ Vulnerability(?y) ^

vulnerabilityName(?x, ?vn1) ^ vulnerabilityName(?y,?vn2) ^ swrlb:notEqual(?vn1, ?vn2) ^

tbox:isSubClassOf(?vc, Vulnerability) ^ abox:hasClass(?x, ?vc) ^ abox:hasClass(?y, ?vc) ^

existInProduct(?x, ?p) ^ existInProduct(?y, ?p) ^ IT\_Product(?p) ^

beExploitedBy(?x, ?z1) ^ beExploitedBy(?y, ?z2) ^ tbox:isSubClassOf(?zc, Attacker) ^

abox:hasClass(?z1, ?zc) ^ abox:hasClass(?z2, ?zc) ^

hasRelatedAttack(?x, ?a1) ^ hasRelatedAttack(?y, ?a2) ^ tbox:isSubClassOf(?ac, Attack)

As similarVulnerability is defined as an “owl:TransitiveProperty” relation, we can then use the OWL transitive property reasoning rule (Figure 7) to help discover a chain of similar vulnerabilities.

owl:TransitiveProperty(?A, ?B) ^ owl:TransitiveProperty(?B, ?C) -> owl:TransitiveProperty(?A, ?C)