

Why Understand Vulnerabilities?

David Shinberg
Internet Research Department
Bell Labs - Lucent Technologies
shinberg@lucent.com



Vulnerability

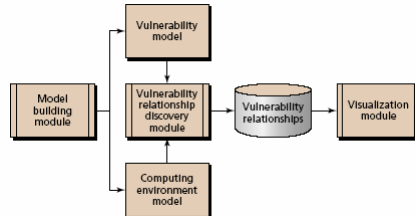
A **vulnerability** is an unplanned system feature that an *intruder* may *exploit*, if he/she can establish certain *preconditions*, to achieve particular *impacts* on that system that violate its *security policy*.

Our goal was to represent vulnerability in a **formal** sense and to learn from that representation.



Modeling

- Model vulnerabilities based on
 - Preconditions
 - Impacts
 - System characteristics
 - User actions



We modeled security-related facts in simple propositional logic, constructed a graph of temporal dependencies among vulnerabilities, and analyzed the resulting graph visualization.

More formal definition of Vulnerability

- $V = f(x)$ where,
- x denotes a subset of X , which describes the current system state, and user behavior.

Exploiting a vulnerability:

Let S = current system state

$V(S) = S'$, where S' is a subset of S .

Vulnerability Modeling Language



- Used CLIPS—C Language integrated production system
 - Modified source to trace track interesting events and generate graph.
- Graph visualization done via GraphViz

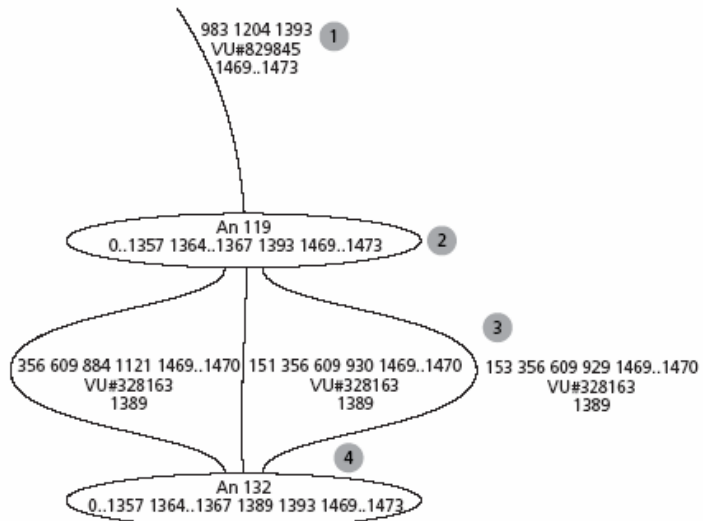
```
(defrule VU#932283
  "Buffer Overflow IE MSHTML control SRC attribute of <EMBED> directive"
  (zonebehavior (action view-malicious-html) (zone ?zone) (user ?user))
  (zonebehavior (action run-malicious-script) (zone ?zone) (user ?user))
  (controlattr (name mshtml-control) (version x)
               (attr scripts) (zone ?zone) (user ?user))
  (controlattr (name mshtml-control) (version x)
               (attr initializes) (zone ?zone) (user ?user))
  (not (patch (name Q317731)))
  =>
  (assert (capability (action execute-unlimited-arbitrary-code) (user ?user)))
  (printout t 'Intruder exploits VU#932283 against user ' ?user crlf))
```

Another Vulnerability Description

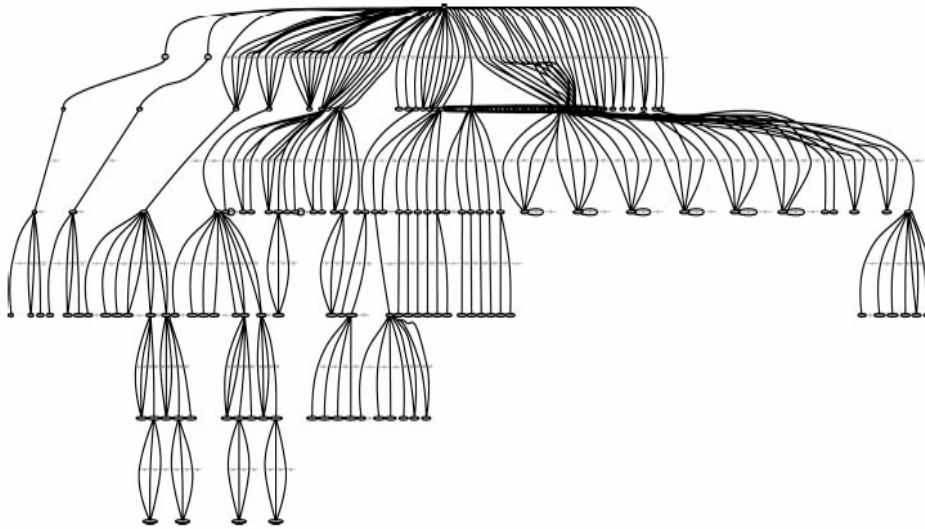


```
(defrule VU#38950 "CVE-2000-0621"
  (or
    (and
      (or
        (software (name internet-explorer) (version 4.01))
        (software (name internet-explorer) (version 5.01)))
      (zonebehavior (action view-malicious-Webpage) (zone ?zone) (user ?user)))
    (and
      (or
        (software (name outlook-express) (version 4.01))
        (software (name outlook-express) (version 5.01)))
      (zonebehavior (action view-malicious-email-message) (zone ?zone) (user ?user))))
  (not (patch (name Q261255)))
  =>
  (assert (capability (action create-arbitrary-file) (user ?user)))
  (printout t 'Intruder exploits VU#38950 against user " ?user crlf))
```

Small Graph



Large Graph 3x12



Conclusion



- Project ended with Bell Labs Technical Journal Publication
 - Formal Modeling of Vulnerability
 - William L. Fithen, Shawn V. Hernan, Paul F. O'Rourke, and David A. Shinberg
 - Bell Labs Technical Journal 8(4), 173–186 (2004)
© 2004 Lucent Technologies Inc. Published by Wiley Periodicals, Inc. Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/bltj.10094

Conclusion



- The approach was validated
 - Identified OLE interactions
 - Improved understanding of vulnerabilities
 - Interesting note: the descriptions did not include the work “**Vulnerability**”
- Needs to be reworked
- Some other projects using similar approaches