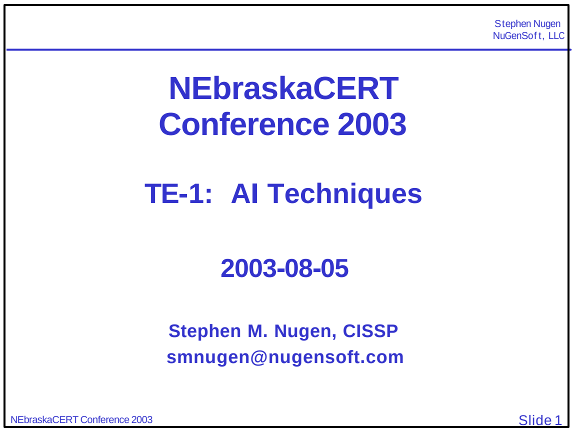
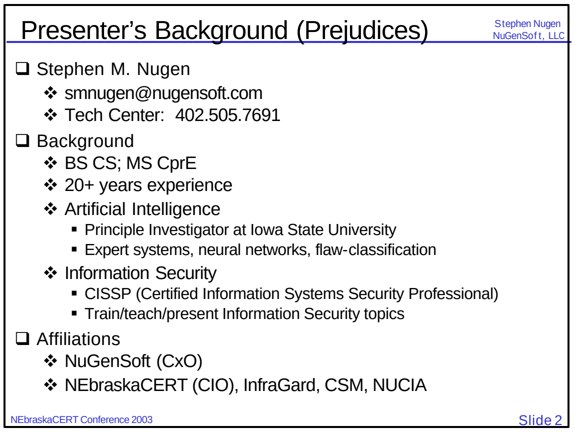


NEbraskaCERT Conference 2003

TE-1: AI Techniques

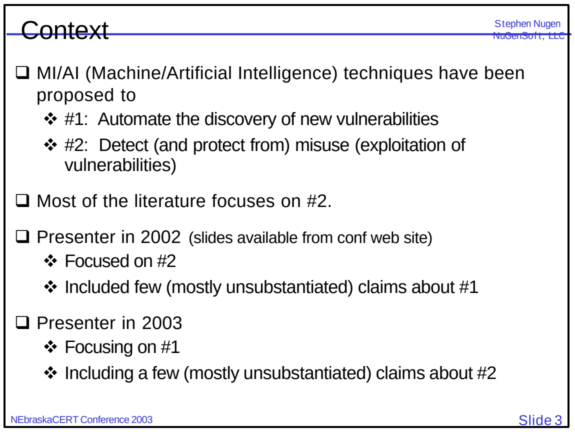
2003-08-05

Stephen M. Nugen, CISSP
smnugen@nugensoft.com



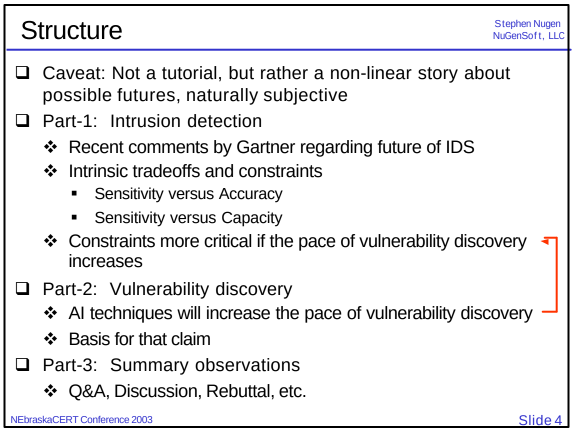
Presenter’s Background (Prejudices)

- ❑ Stephen M. Nugen
  - ❖ smnugen@nugensoft.com
  - ❖ Tech Center: 402.505.7691
- ❑ Background
  - ❖ BS CS; MS CprE
  - ❖ 20+ years experience
  - ❖ Artificial Intelligence
    - Principle Investigator at Iowa State University
    - Expert systems, neural networks, flaw-classification
  - ❖ Information Security
    - CISSP (Certified Information Systems Security Professional)
    - Train/teach/present Information Security topics
- ❑ Affiliations
  - ❖ NuGenSoft (CxO)
  - ❖ NEbraskaCERT (CIO), InfraGard, CSM, NUCIA



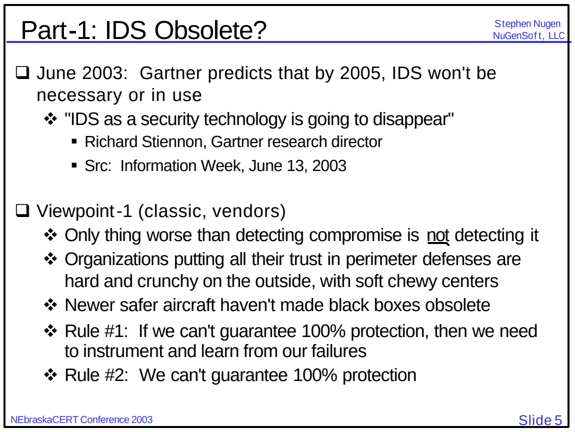
Context

- ❑ MI/AI (Machine/Artificial Intelligence) techniques have been proposed to
  - ❖ #1: Automate the discovery of new vulnerabilities
  - ❖ #2: Detect (and protect from) misuse (exploitation of vulnerabilities)
- ❑ Most of the literature focuses on #2.
- ❑ Presenter in 2002 (slides available from conf web site)
  - ❖ Focused on #2
  - ❖ Included few (mostly unsubstantiated) claims about #1
- ❑ Presenter in 2003
  - ❖ Focusing on #1
  - ❖ Including a few (mostly unsubstantiated) claims about #2



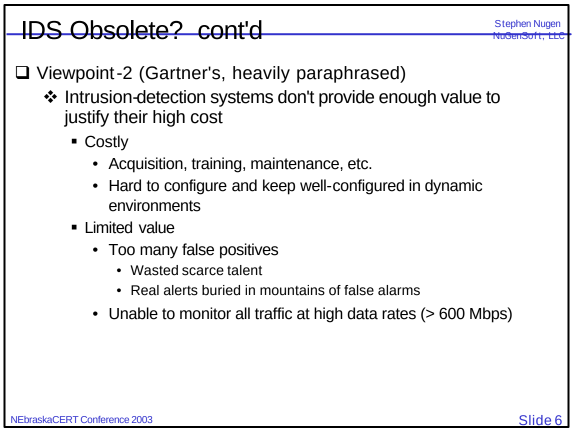
Structure

- ❑ Caveat: Not a tutorial, but rather a non-linear story about possible futures, naturally subjective
- ❑ Part-1: Intrusion detection
  - ❖ Recent comments by Gartner regarding future of IDS
  - ❖ Intrinsic tradeoffs and constraints
    - ▪ Sensitivity versus Accuracy
    - ▪ Sensitivity versus Capacity
  - ❖ Constraints more critical if the pace of vulnerability discovery increases
- ❑ Part-2: Vulnerability discovery
  - ❖ AI techniques will increase the pace of vulnerability discovery
  - ❖ Basis for that claim
- ❑ Part-3: Summary observations
  - ❖ Q&A, Discussion, Rebuttal, etc.



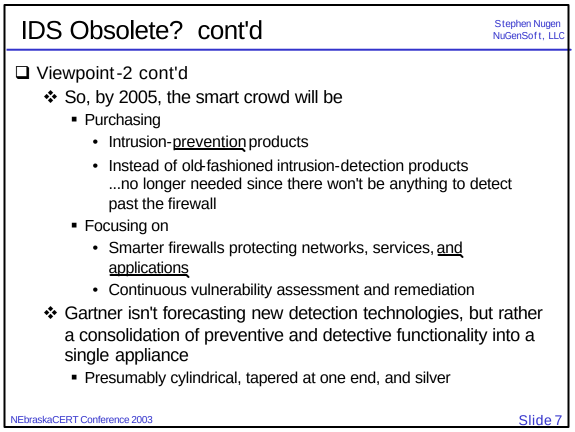
Part-1: IDS Obsolete?

- ❑ June 2003: Gartner predicts that by 2005, IDS won't be necessary or in use
  - ❖ "IDS as a security technology is going to disappear"
    - ▪ Richard Stiennon, Gartner research director
    - ▪ Src: Information Week, June 13, 2003
- ❑ Viewpoint-1 (classic, vendors)
  - ❖ Only thing worse than detecting compromise is <u>not</u> detecting it
  - ❖ Organizations putting all their trust in perimeter defenses are hard and crunchy on the outside, with soft chewy centers
  - ❖ Newer safer aircraft haven't made black boxes obsolete
  - ❖ Rule #1: If we can't guarantee 100% protection, then we need to instrument and learn from our failures
  - ❖ Rule #2: We can't guarantee 100% protection



IDS Obsolete? cont'd

- ❑ Viewpoint-2 (Gartner's, heavily paraphrased)
  - ❖ Intrusion-detection systems don't provide enough value to justify their high cost
    - ▪ Costly
      - Acquisition, training, maintenance, etc.
      - Hard to configure and keep well-configured in dynamic environments
    - ▪ Limited value
      - Too many false positives
        - Wasted scarce talent
        - Real alerts buried in mountains of false alarms
      - Unable to monitor all traffic at high data rates (> 600 Mbps)

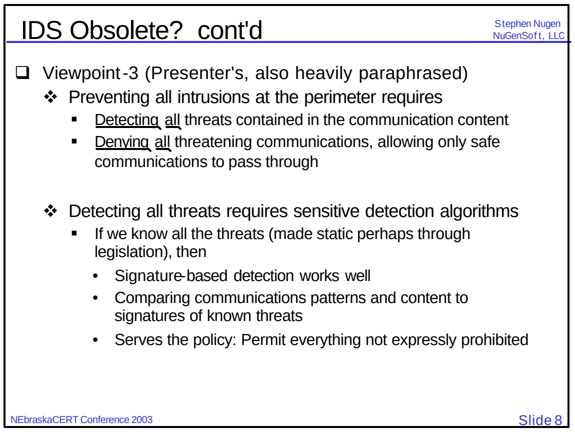


IDS Obsolete? cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Viewpoint-2 cont'd
  - ❖ So, by 2005, the smart crowd will be
    - ▪ Purchasing
      - Intrusion-prevention products
      - Instead of old-fashioned intrusion-detection products ...no longer needed since there won't be anything to detect past the firewall
    - ▪ Focusing on
      - Smarter firewalls protecting networks, services, and applications
      - Continuous vulnerability assessment and remediation
  - ❖ Gartner isn't forecasting new detection technologies, but rather a consolidation of preventive and detective functionality into a single appliance
    - ▪ Presumably cylindrical, tapered at one end, and silver

NEbraskaCERT Conference 2003 Slide 7

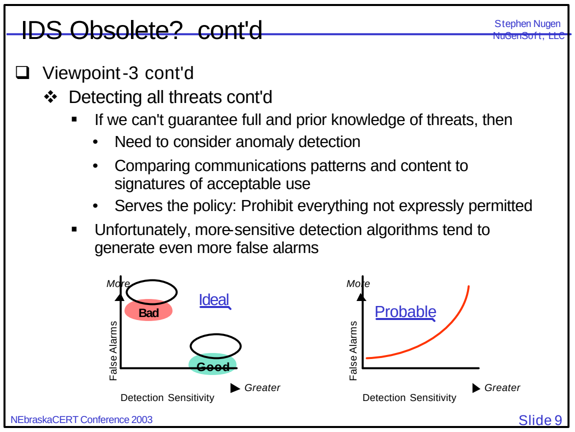


IDS Obsolete? cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Viewpoint-3 (Presenter's, also heavily paraphrased)
  - ❖ Preventing all intrusions at the perimeter requires
    - ▪ Detecting all threats contained in the communication content
    - ▪ Denying all threatening communications, allowing only safe communications to pass through
  - ❖ Detecting all threats requires sensitive detection algorithms
    - ▪ If we know all the threats (made static perhaps through legislation), then
      - Signature-based detection works well
      - Comparing communications patterns and content to signatures of known threats
      - Serves the policy: Permit everything not expressly prohibited

NEbraskaCERT Conference 2003 Slide 8

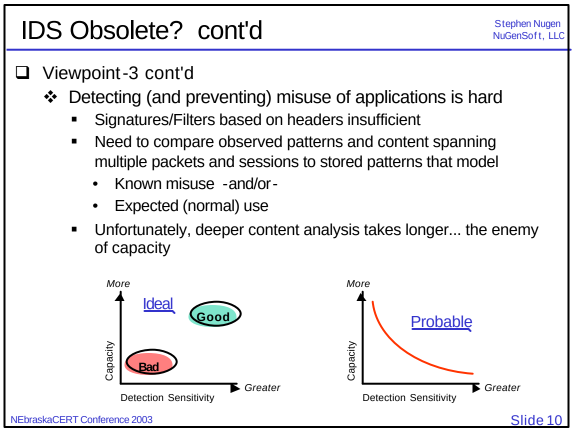


IDS Obsolete? cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Viewpoint-3 cont'd
  - ❖ Detecting all threats cont'd
    - ▪ If we can't guarantee full and prior knowledge of threats, then
      - Need to consider anomaly detection
      - Comparing communications patterns and content to signatures of acceptable use
      - Serves the policy: Prohibit everything not expressly permitted
    - ▪ Unfortunately, more-sensitive detection algorithms tend to generate even more false alarms

NEbraskaCERT Conference 2003 Slide 9



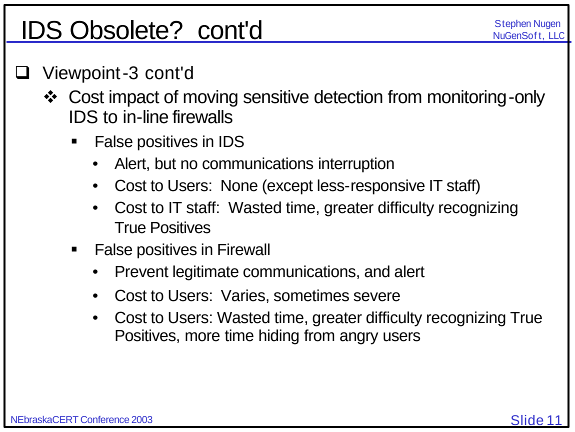
IDS Obsolete? cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Viewpoint-3 cont'd
  - ❖ Detecting (and preventing) misuse of applications is hard
    - ▪ Signatures/Filters based on headers insufficient
    - ▪ Need to compare observed patterns and content spanning multiple packets and sessions to stored patterns that model
      - Known misuse -and/or-
      - Expected (normal) use
    - ▪ Unfortunately, deeper content analysis takes longer... the enemy of capacity

NEbraskaCERT Conference 2003

Slide 10



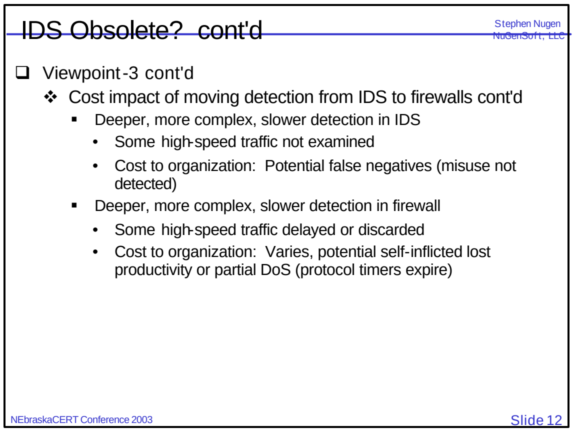
IDS Obsolete? cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Viewpoint-3 cont'd
  - ❖ Cost impact of moving sensitive detection from monitoring-only IDS to in-line firewalls
    - ▪ False positives in IDS
      - Alert, but no communications interruption
      - Cost to Users: None (except less-responsive IT staff)
      - Cost to IT staff: Wasted time, greater difficulty recognizing True Positives
    - ▪ False positives in Firewall
      - Prevent legitimate communications, and alert
      - Cost to Users: Varies, sometimes severe
      - Cost to Users: Wasted time, greater difficulty recognizing True Positives, more time hiding from angry users

NEbraskaCERT Conference 2003

Slide 11



IDS Obsolete? cont'd

Stephen Nugen
NuGenSoft, LLC

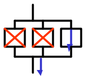
- ❑ Viewpoint-3 cont'd
  - ❖ Cost impact of moving detection from IDS to firewalls cont'd
    - ▪ Deeper, more complex, slower detection in IDS
      - Some high-speed traffic not examined
      - Cost to organization: Potential false negatives (misuse not detected)
    - ▪ Deeper, more complex, slower detection in firewall
      - Some high-speed traffic delayed or discarded
      - Cost to organization: Varies, potential self-inflicted lost productivity or partial DoS (protocol timers expire)

NEbraskaCERT Conference 2003

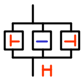
Slide 12

Claim-A

- ❑ Regardless of where the detection function resides, utility depends on
  - ❖ Capacity (how much of the traffic is examined) ... primarily function of speed since parallel processing
    - ▪ Not feasible for in-line firewalls



    - ▪ Not feasible for IDS when hostile activity spans multiple sessions



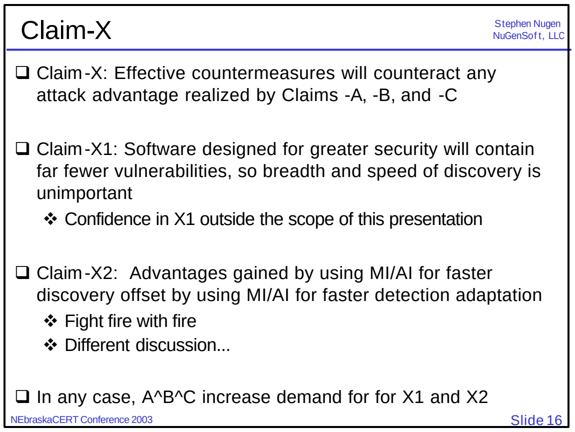
  - ❖ Accuracy, permitting no more than acceptable number of confidence-lowering
    - ▪ False positives (legitimate content detected as hostile)
    - ▪ False negatives (hostile content not detected as hostile)
  - ❖ Confidence in Claim-A is near-universal.

Claim-B cont'd

- ❑ Detection utility also depends on:
  - ❖ How quickly our detection methods and implementations adapt/evolve, relative to speed at which attacks evolve
    - ▪ Quickness of adaptation directly impacts accuracy
  - ❖ How efficiently our detection methods and implementations adapt/evolve in response to evolving attacks
    - ▪ Efficiency of adaptation directly impacts capacity
- ❑ Confidence in Claim-B less universal, but growing in response to
  - ❖ Multi-vector attacks like NIMDA
  - ❖ Evolving malware like SoBig
  - ❖ Quick-to-market exploits like ShadowCode and RPC/DCOM exploits from Xfocus and Metasploit

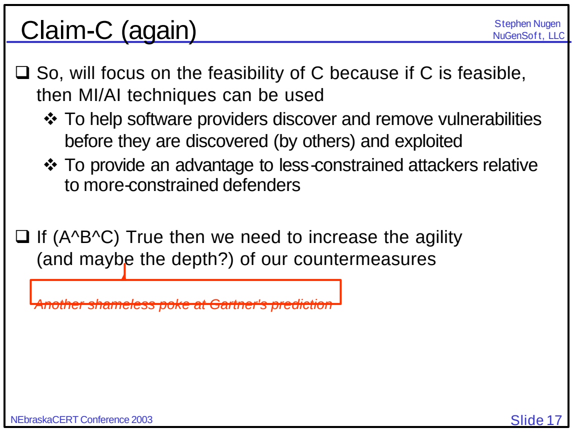
Claim-C

- ❑ MI/AI techniques can and will be used to assist in the discovery of new vulnerabilities in commercial and custom software
  - ❖ Increasing the number of exploitable vulnerabilities
  - ❖ Increasing the speed at which attacks can evolve
- ❑ Importance: If (Claim-A True <u>and</u> Claim-B True <u>and</u> Claim-C True <u>and</u> Claim-X False) Then
  - ▪ Speed at which attacks evolve will increase relative to speed of detection adaptations
  - ▪ More vulnerabilities and corresponding exploits increase the difficulty of vendors
    - Updating misuse signatures
    - Patching the vulnerability
  - ▪ More attacks succeed



Claim-X

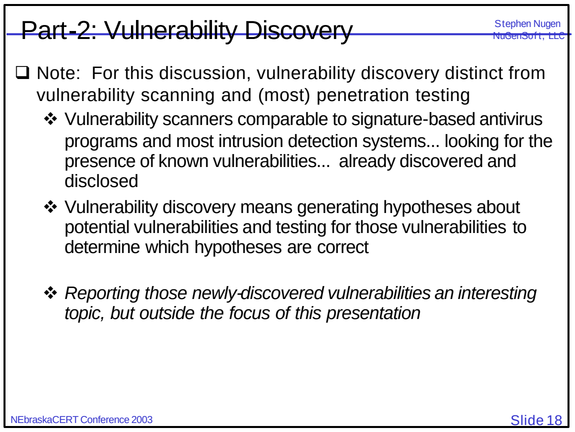
- Claim-X: Effective countermeasures will counteract any attack advantage realized by Claims -A, -B, and -C
- Claim-X1: Software designed for greater security will contain far fewer vulnerabilities, so breadth and speed of discovery is unimportant
  - Confidence in X1 outside the scope of this presentation
- Claim-X2: Advantages gained by using MI/AI for faster discovery offset by using MI/AI for faster detection adaptation
  - Fight fire with fire
  - Different discussion...
- In any case, A^B^C increase demand for for X1 and X2



Claim-C (again)

- So, will focus on the feasibility of C because if C is feasible, then MI/AI techniques can be used
  - To help software providers discover and remove vulnerabilities before they are discovered (by others) and exploited
  - To provide an advantage to less-constrained attackers relative to more-constrained defenders
- If (A^B^C) True then we need to increase the agility (and maybe the depth?) of our countermeasures

Another shameless poke at Gartner's prediction



Part-2: Vulnerability Discovery

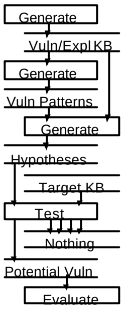
- Note: For this discussion, vulnerability discovery distinct from vulnerability scanning and (most) penetration testing
  - Vulnerability scanners comparable to signature-based antivirus programs and most intrusion detection systems... looking for the presence of known vulnerabilities... already discovered and disclosed
  - Vulnerability discovery means generating hypotheses about potential vulnerabilities and testing for those vulnerabilities to determine which hypotheses are correct
  - *Reporting those newly-discovered vulnerabilities an interesting topic, but outside the focus of this presentation*

Basis For Claim-C

❑ Approach for discussing the feasibility of Claim-C

- ❖ Break large claim into smaller pieces
- ❖ Show larger Claim-C feasible by showing all the (required) pieces are feasible

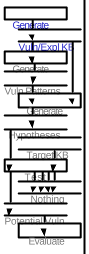
❑ *Note: This discussion is notional, conjecture... not a discussion of anyone's specific architecture or any proposed architecture*



Vulnerability KB

❑ Content

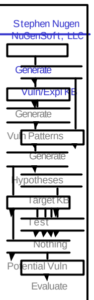
- ❖ Known, reported vulnerabilities and exploits
- ❖ Example
  - ▪ Preconditions
    - • Access Type (e.g., External, Internal, Inserted)
    - • Privilege Level (e.g., Anonymous, Auth-User, Root)
    - • Operating Environment (e.g., Vendor, Software version, etc.)
    - • Predecessors (for chained exploits)
  - ▪ Operations
    - • Exploit Method (e.g., Malformed Input, Impersonation, etc.)
    - • Known Exploits (the messy details)
    - • Comments (e.g., weakness associated with vulnerability, when announced/mitigated, etc.)



Vulnerability KB cont'd

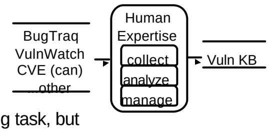
❑ Content cont'd

- ❖ Example cont'd
  - ▪ Postconditions
    - • Result (e.g., Remote Control, DoS, File Access, etc.)
    - • Successors (for chained exploits)
  - ▪ Mitigations
    - • Operational (e.g., port filtering, terminate service, etc.)
    - • Updates (e.g., patches, new software versions, etc.)
  - ▪ Other...



Vulnerability KB cont'd

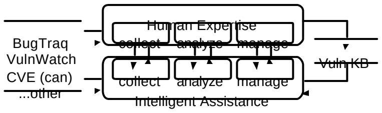
- □ Generating Vuln KB



  - ❖ Rather tedious, continuing task, but
    - Effort can be distributed over multiple experts
    - Results can be shared
  - ❖ Technical challenges include
    - Analyzing and structuring the information for pattern development and generating hypotheses
  - ❖ Management challenges include
    - Agreeing on common definitions, data exchange formats, etc.
    - Coordinating efforts of multiple experts
    - Possible legal constraints on full-disclosure

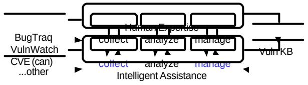
Vulnerability KB cont'd

- □ Feasibility
  - ❖ Relief: No strict minimums thresholds for scope or performance
  - ❖ Human-feasible: Yes
    - With or without MI/AI technologies
    - (Partial) Examples: Vendors and open source communities defining vulnerability signatures; CVE
- □ MI/AI opportunities: Augment human expertise with intelligent assistance



Vulnerability KB cont'd

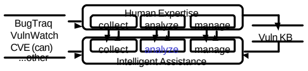
- □ MI/AI opportunities cont'd



  - ❖ Distributed agents may be employed to
    - Collect information... avoiding duplication
      - Cooperating agents, resolving duplicates between themselves with direct communications or shared blackboard
      - Hierarchical control
    - Help coordinate collection and analysis tasks between different human experts
      - Example
        - Raw Info -> three suitable experts
        - Wait for at least two responses (analyses)... Nag as required
        - Send responses to suitable moderator (to resolve differences)

Vulnerability KB cont'd

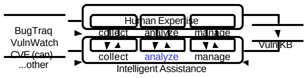
❑ MI/AI opportunities cont'd



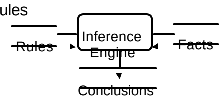
- ❖ Natural Language Processing (NLP) techniques may be used for first-level content parsing
  - ▪ Some sources easier to parse than others...
  - ▪ First-level might be sufficient to recognize
    - Duplicates
    - Partial-Duplicates, distinguishing just the information added or changed

Vulnerability KB cont'd

❑ MI/AI opportunities cont'd



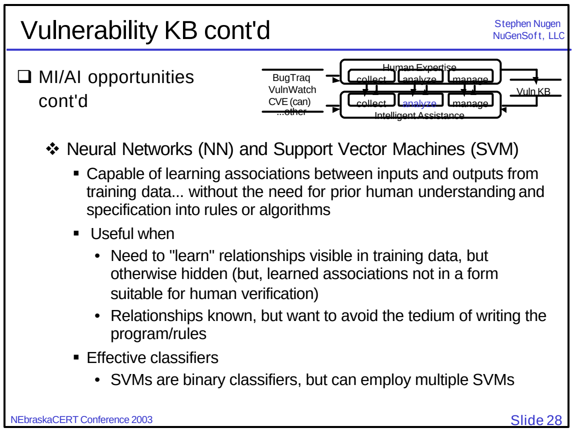
- ❖ Expert system
  - ▪ Rules may be employed to express heuristic knowledge (easier than code to review/change)
  - ▪ Inference engines evaluate (fire) the rules (goal-directed, forward-chaining)
  - ▪ Conclusions may
    - Invoke a new action
    - Increase belief or disbelief in a specific hypothesis (beliefs accumulate)



Vulnerability KB cont'd

❑ MI/AI opportunities cont'd

- ❖ Expert system cont'd
  - ▪ Example
    - IF (MS Sec Bulletin) AND (Vuln-Text includes the phrase "Microsoft thanks <X> "for reporting this issue to us and working with us to protect customers")
    - THEN (Collection-Task = Collect more information from <X>)
  - ▪ Example
    - IF (Vuln-Text includes the phrase "run code of attacker's choice")
    - THEN
      - Assert strong belief (Post.Expl-Result, DoS,,)
      - Assert strong belief(Post.Expl-Result, Rem-Control,,)
      - Assert moderate belief (Post.Expl-Result, File-Access,RWX)
      - Assert moderate belief (Post.Succ, *multiple,,)



Vulnerability KB cont'd

- ❑ MI/AI opportunities cont'd

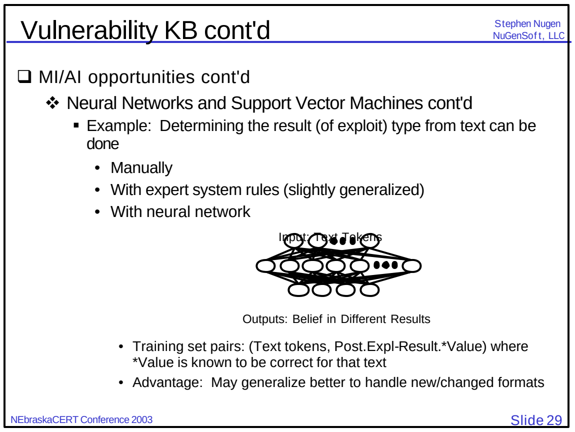
BugTraq
VulnWatch
CVE (can)
...other

Human Expertise
collect
analyze
manage

collect
analyze
manage
Intelligent Assistance

Vuln KB

- ❖ Neural Networks (NN) and Support Vector Machines (SVM)
  - ▪ Capable of learning associations between inputs and outputs from training data... without the need for prior human understanding and specification into rules or algorithms
  - ▪ Useful when
    - Need to "learn" relationships visible in training data, but otherwise hidden (but, learned associations not in a form suitable for human verification)
    - Relationships known, but want to avoid the tedium of writing the program/rules
  - ▪ Effective classifiers
    - SVMs are binary classifiers, but can employ multiple SVMs



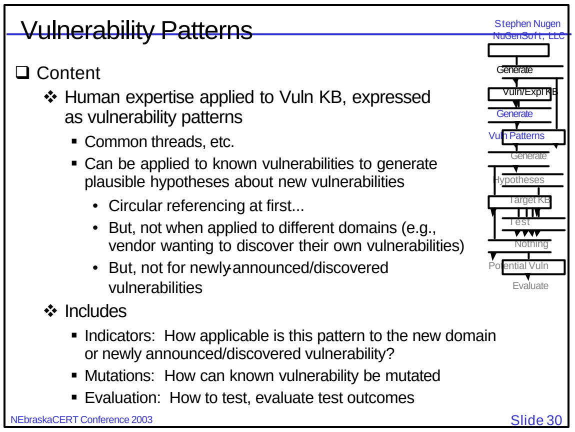
Vulnerability KB cont'd

- ❑ MI/AI opportunities cont'd
  - ❖ Neural Networks and Support Vector Machines cont'd
    - ▪ Example: Determining the result (of exploit) type from text can be done
      - Manually
      - With expert system rules (slightly generalized)
      - With neural network

Inputs: Text Tokens

Outputs: Belief in Different Results

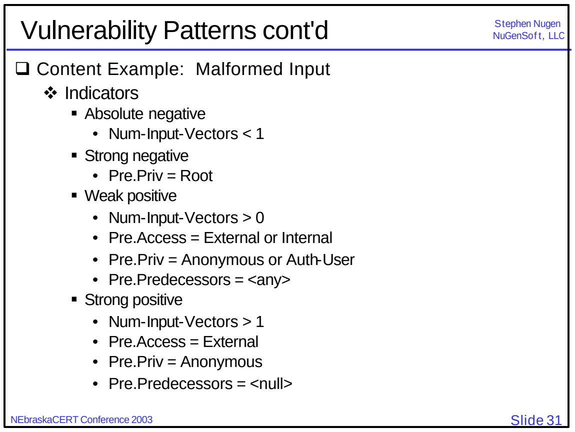
- Training set pairs: (Text tokens, Post.Expl-Result.*Value) where *Value is known to be correct for that text
- Advantage: May generalize better to handle new/changed formats



Vulnerability Patterns

Generate
Vuln/Expl KB
Generate
Vuln Patterns
Generate
Hypotheses
Target KB
Test
Nothing
Potential Vuln
Evaluate

- ❑ Content
  - ❖ Human expertise applied to Vuln KB, expressed as vulnerability patterns
    - ▪ Common threads, etc.
    - ▪ Can be applied to known vulnerabilities to generate plausible hypotheses about new vulnerabilities
      - Circular referencing at first...
      - But, not when applied to different domains (e.g., vendor wanting to discover their own vulnerabilities)
      - But, not for newly-announced/discovered vulnerabilities
  - ❖ Includes
    - ▪ Indicators: How applicable is this pattern to the new domain or newly announced/discovered vulnerability?
    - ▪ Mutations: How can known vulnerability be mutated
    - ▪ Evaluation: How to test, evaluate test outcomes



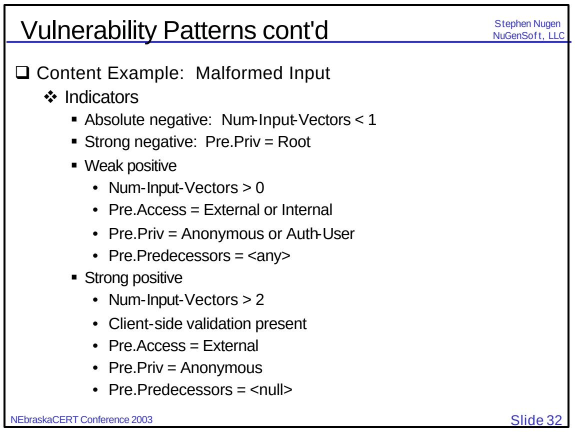
Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

❑ Content Example: Malformed Input
- ❖ Indicators
  - Absolute negative
    - Num-Input-Vectors < 1
  - Strong negative
    - Pre.Priv = Root
  - Weak positive
    - Num-Input-Vectors > 0
    - Pre.Access = External or Internal
    - Pre.Priv = Anonymous or Auth-User
    - Pre.Predecessors = <any>
  - Strong positive
    - Num-Input-Vectors > 1
    - Pre.Access = External
    - Pre.Priv = Anonymous
    - Pre.Predecessors = <null>

NEbraskaCERT Conference 2003

Slide 31



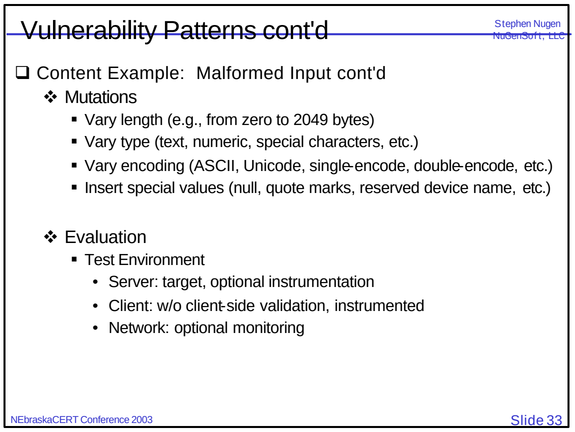
Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

❑ Content Example: Malformed Input
- ❖ Indicators
  - Absolute negative: Num-Input-Vectors < 1
  - Strong negative: Pre.Priv = Root
  - Weak positive
    - Num-Input-Vectors > 0
    - Pre.Access = External or Internal
    - Pre.Priv = Anonymous or Auth-User
    - Pre.Predecessors = <any>
  - Strong positive
    - Num-Input-Vectors > 2
    - Client-side validation present
    - Pre.Access = External
    - Pre.Priv = Anonymous
    - Pre.Predecessors = <null>

NEbraskaCERT Conference 2003

Slide 32



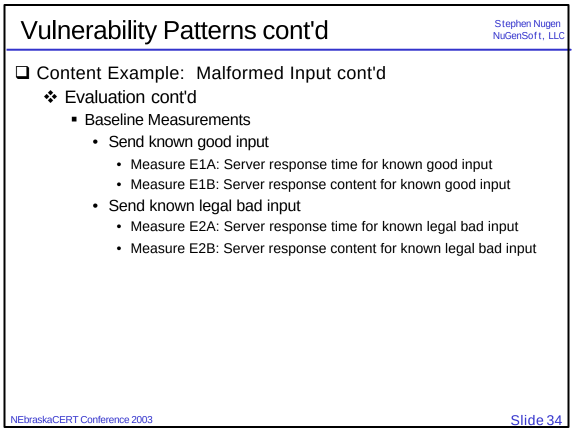
Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

❑ Content Example: Malformed Input cont'd
- ❖ Mutations
  - Vary length (e.g., from zero to 2049 bytes)
  - Vary type (text, numeric, special characters, etc.)
  - Vary encoding (ASCII, Unicode, single-encode, double-encode, etc.)
  - Insert special values (null, quote marks, reserved device name, etc.)
- ❖ Evaluation
  - Test Environment
    - Server: target, optional instrumentation
    - Client: w/o client-side validation, instrumented
    - Network: optional monitoring

NEbraskaCERT Conference 2003

Slide 33



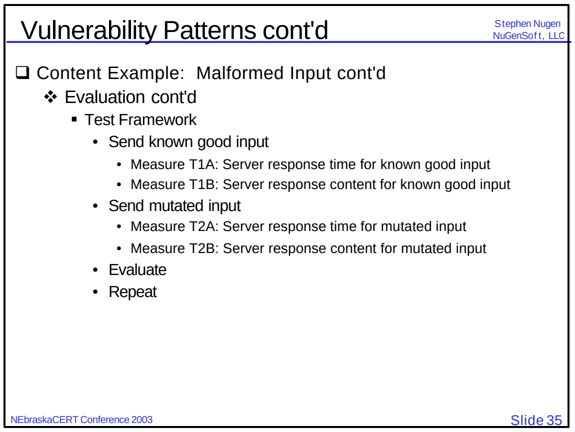
Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Content Example: Malformed Input cont'd
  - ❖ Evaluation cont'd
    - ▪ Baseline Measurements
      - Send known good input
        - Measure E1A: Server response time for known good input
        - Measure E1B: Server response content for known good input
      - Send known legal bad input
        - Measure E2A: Server response time for known legal bad input
        - Measure E2B: Server response content for known legal bad input

NEbraskaCERT Conference 2003

Slide 34



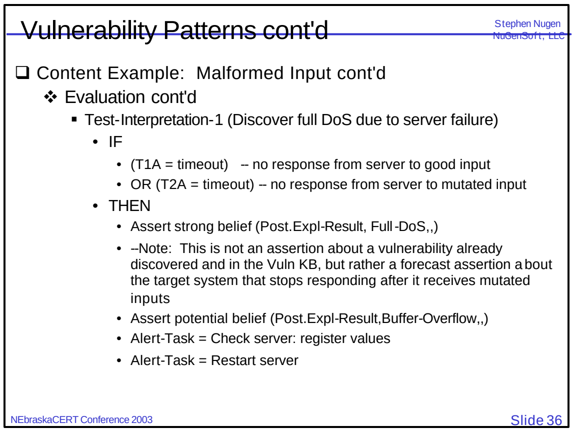
Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Content Example: Malformed Input cont'd
  - ❖ Evaluation cont'd
    - ▪ Test Framework
      - Send known good input
        - Measure T1A: Server response time for known good input
        - Measure T1B: Server response content for known good input
      - Send mutated input
        - Measure T2A: Server response time for mutated input
        - Measure T2B: Server response content for mutated input
      - Evaluate
      - Repeat

NEbraskaCERT Conference 2003

Slide 35



Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

- ❑ Content Example: Malformed Input cont'd
  - ❖ Evaluation cont'd
    - ▪ Test-Interpretation-1 (Discover full DoS due to server failure)
      - IF
        - (T1A = timeout)   -- no response from server to good input
        - OR (T2A = timeout) -- no response from server to mutated input
      - THEN
        - Assert strong belief (Post.Expl-Result, Full-DoS,,)
        - --Note: This is not an assertion about a vulnerability already discovered and in the Vuln KB, but rather a forecast assertion about the target system that stops responding after it receives mutated inputs
        - Assert potential belief (Post.Expl-Result,Buffer-Overflow,,)
        - Alert-Task = Check server: register values
        - Alert-Task = Restart server

NEbraskaCERT Conference 2003

Slide 36

Vulnerability Patterns cont'd

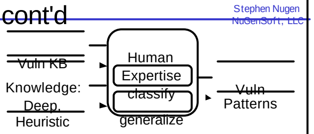
- ❑ Content Example: Malformed Input cont'd
  - ❖ Evaluation cont'd
    - ▪ Test-Interpretation-2 (Discover partial DoS due to Server error/exception processing)
      - IF
        - (T1A >> E1A) – server has slowed down, even for good inputs
        - OR (T2A >> E2A) – server responds slower to mutated inputs
      - THEN
        - Assert moderate belief (Post.Expl-Result, Partial-DoS,,)
        - Continue

Vulnerability Patterns cont'd

- ❑ Content Example: Partial DoS -> Full DoS
  - ❖ Indicators
    - ▪ Absolute negative
      - (Belief (Post.Expl-Result, Partial-DoS,,) < unknown)
    - ▪ Strong negative
      - (Belief (Post.Expl-Result, Partial-DoS,,) = unknown)
    - ▪ Weak positive
      - Belief (Post.Expl-Result, Partial-DoS,,) > unknown
    - ▪ Strong positive
      - Belief (Post.Expl-Result, Partial-DoS,,) > weak
  - ❖ Mutations
    - ▪ Vary single-client volume (just blast, without waiting for response)
    - ▪ Vary number of clients (use multiple clients for discover DDoS)
  - ❖ Evaluation... similar to previous example

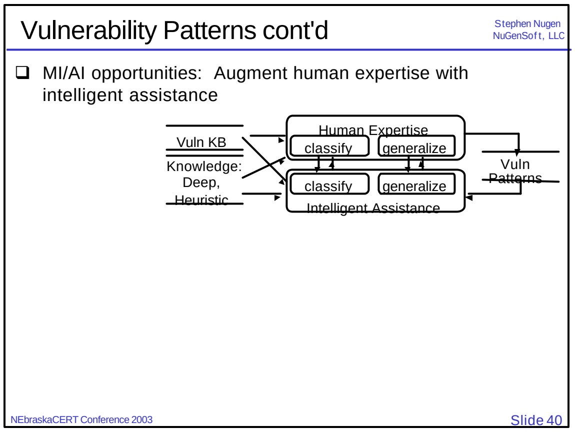
Vulnerability Patterns cont'd

- ❑ Generating Vuln Patterns



Less tedious, less continual than populating the Vuln KB
- ▪ Efforts distributable, results shareable

- ❑ Feasibility
  - ❖ Relief: No strict minimums thresholds for scope or performance
  - ❖ Human-feasible: Yes
    - ▪ With fewer, but more skilled, human experts
    - ▪ With or without MI/AI technologies
    - ▪ Examples: White-hat and Black-hat hackers



Vulnerability Patterns cont'd

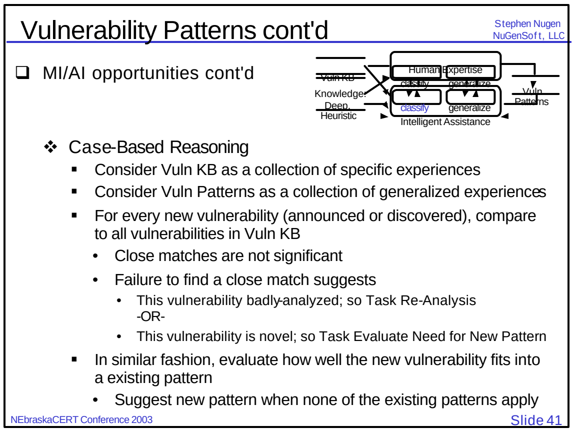
Stephen Nugen
NuGenSoft, LLC

❑ MI/AI opportunities: Augment human expertise with intelligent assistance

Vuln KB
Knowledge: Deep, Heuristic
Human Expertise
classify
generalize
classify
generalize
Intelligent Assistance
Vuln Patterns

NEbraskaCERT Conference 2003

Slide 40



Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

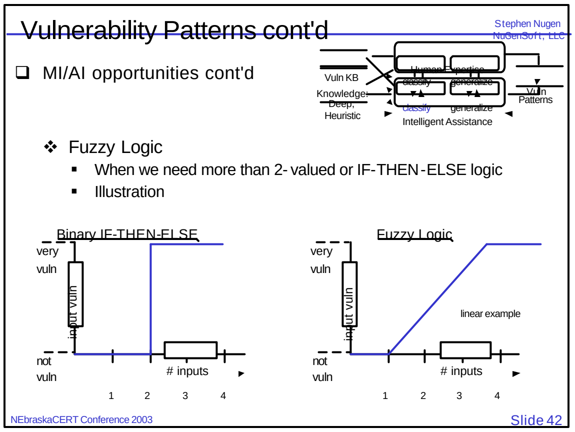
❑ MI/AI opportunities cont'd

Vuln KB
Knowledge: Deep, Heuristic
Human Expertise
classify
generalize
classify
generalize
Intelligent Assistance
Vuln Patterns

- ❖ Case-Based Reasoning
  - ▪ Consider Vuln KB as a collection of specific experiences
  - ▪ Consider Vuln Patterns as a collection of generalized experiences
  - ▪ For every new vulnerability (announced or discovered), compare to all vulnerabilities in Vuln KB
    - Close matches are not significant
    - Failure to find a close match suggests
      - This vulnerability badly-analyzed; so Task Re-Analysis -OR-
      - This vulnerability is novel; so Task Evaluate Need for New Pattern
  - ▪ In similar fashion, evaluate how well the new vulnerability fits into a existing pattern
    - Suggest new pattern when none of the existing patterns apply

NEbraskaCERT Conference 2003

Slide 41



Vulnerability Patterns cont'd

Stephen Nugen
NuGenSoft, LLC

❑ MI/AI opportunities cont'd

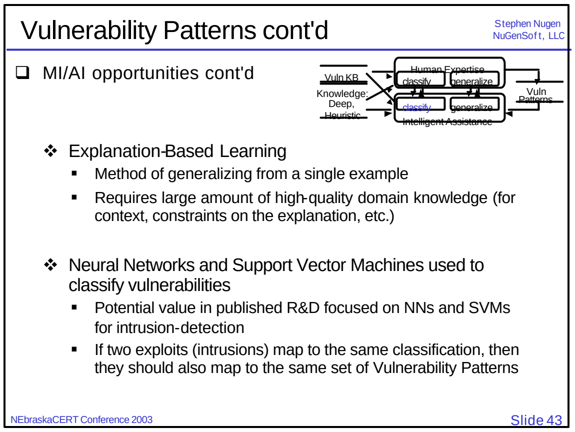
Vuln KB
Knowledge: Deep, Heuristic
Human Expertise
classify
generalize
classify
generalize
Intelligent Assistance
Vuln Patterns

- ❖ Fuzzy Logic
  - ▪ When we need more than 2-valued or IF-THEN-ELSE logic
  - ▪ Illustration

Binary IF-THEN-ELSE
very vuln
input vuln
not vuln
# inputs
1
2
3
4
Fuzzy Logic
very vuln
input vuln
linear example
not vuln
# inputs
1
2
3
4

NEbraskaCERT Conference 2003

Slide 42



Vulnerability Patterns cont'd

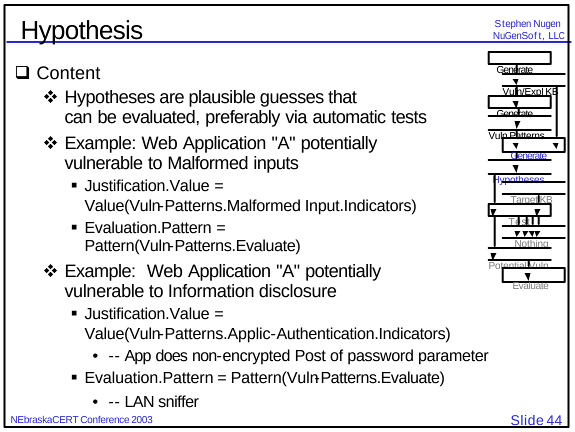
Stephen Nugen
NuGenSoft, LLC

❑ MI/AI opportunities cont'd

Vuln KB
Knowledge: Deep, Heuristic
Human Expertise
classify
generalize
Vuln Patterns
classify
generalize
Intelligent Assistance

- ❖ Explanation-Based Learning
  - Method of generalizing from a single example
  - Requires large amount of high-quality domain knowledge (for context, constraints on the explanation, etc.)
- ❖ Neural Networks and Support Vector Machines used to classify vulnerabilities
  - Potential value in published R&D focused on NNs and SVMs for intrusion-detection
  - If two exploits (intrusions) map to the same classification, then they should also map to the same set of Vulnerability Patterns

NEbraskaCERT Conference 2003 Slide 43



Hypothesis

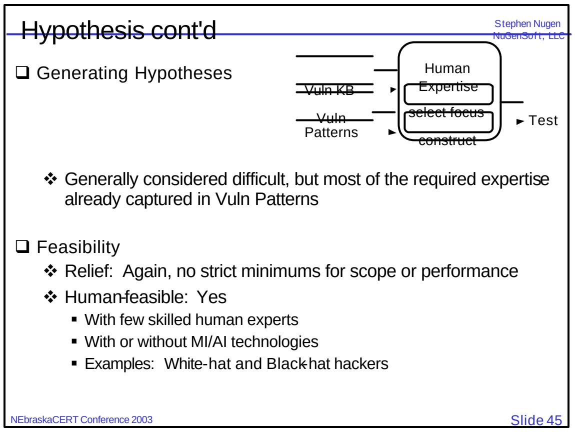
Stephen Nugen
NuGenSoft, LLC

Generate
Vuln/Expl KB
Generate
Vuln Patterns
Generate
Hypotheses
Target KB
Test
Nothing
Potential Vuln
Evaluate

❑ Content

- ❖ Hypotheses are plausible guesses that can be evaluated, preferably via automatic tests
- ❖ Example: Web Application "A" potentially vulnerable to Malformed inputs
  - Justification.Value = Value(Vuln-Patterns.Malformed Input.Indicators)
  - Evaluation.Pattern = Pattern(Vuln-Patterns.Evaluate)
- ❖ Example: Web Application "A" potentially vulnerable to Information disclosure
  - Justification.Value = Value(Vuln-Patterns.Applic-Authentication.Indicators)
    - -- App does non-encrypted Post of password parameter
  - Evaluation.Pattern = Pattern(Vuln-Patterns.Evaluate)
    - -- LAN sniffer

NEbraskaCERT Conference 2003 Slide 44



Hypothesis cont'd

Stephen Nugen
NuGenSoft, LLC

❑ Generating Hypotheses

Vuln KB
Vuln Patterns
Human Expertise
select focus
construct
Test

- ❖ Generally considered difficult, but most of the required expertise already captured in Vuln Patterns

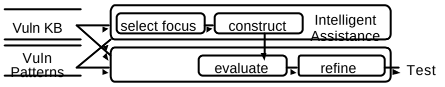
❑ Feasibility

- ❖ Relief: Again, no strict minimums for scope or performance
- ❖ Human-feasible: Yes
  - With few skilled human experts
  - With or without MI/AI technologies
  - Examples: White-hat and Black-hat hackers

NEbraskaCERT Conference 2003 Slide 45

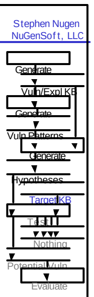
Hypothesis cont'd

- ❑ MI/AI opportunities: Augment human expertise by reassigning some of the iterative tasks to software
  - ❖ Advantage comes from use abundant machine cycles to test a very wide range of hypotheses (and mutations)
  - ❖ Use relatively simple programs to iteratively generate the different combinations of hypotheses
    - Keep track of the justifications for human expert confirmation



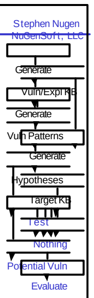
Target KB

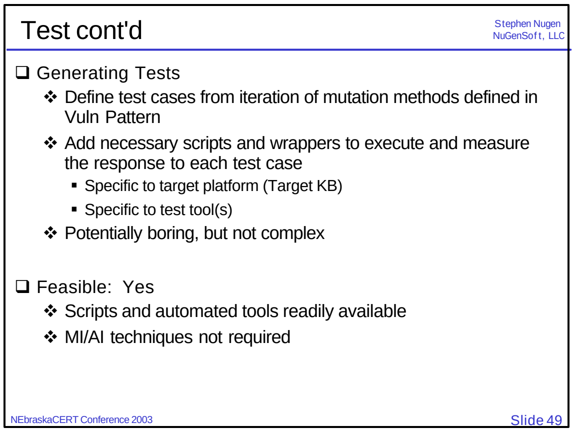
- ❑ Content
  - ❖ Meta-knowledge describing the environment vulnerability discovery is focused on
  - ❖ Example
    - Target Attributes
      - IP address,
      - URLs,
      - Post Form parameters
    - Constraints
- ❑ Feasible: Yes, low-risk



Test

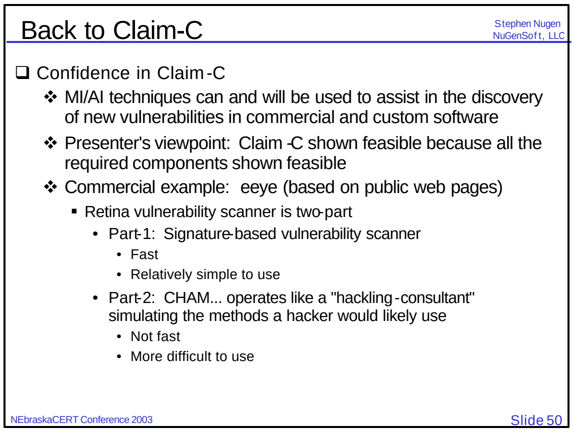
- ❑ Content
  - ❖ For each Hypothesis, generate test cases that reflect all or a significant subset of all possible mutations as defined in the applicable Vuln Pattern
  - ❖ Example for one Hypothesis
    - Assume
      - Known good input: "A"
      - Mutation-Method-1: Vary input-1 length: [1 - 1025]
      - Mutation-Method-2: Insert special characters: [<null>, <%>, <'>, <–>]
    - Test-1: Input = "A"
    - Test-2: Input = "AA"
    - Test-1025: Input = "A.....A" (1025 bytes)
    - Test-1026 = <null>
    - Test-1027 = "A<null>"
    - ...and so forth





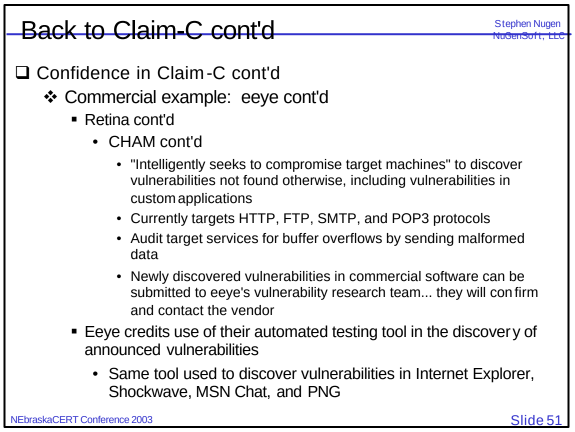
Test cont'd

- ❑ Generating Tests
  - ❖ Define test cases from iteration of mutation methods defined in Vuln Pattern
  - ❖ Add necessary scripts and wrappers to execute and measure the response to each test case
    - ▪ Specific to target platform (Target KB)
    - ▪ Specific to test tool(s)
  - ❖ Potentially boring, but not complex
- ❑ Feasible: Yes
  - ❖ Scripts and automated tools readily available
  - ❖ MI/AI techniques not required



Back to Claim-C

- ❑ Confidence in Claim-C
  - ❖ MI/AI techniques can and will be used to assist in the discovery of new vulnerabilities in commercial and custom software
  - ❖ Presenter's viewpoint: Claim-C shown feasible because all the required components shown feasible
  - ❖ Commercial example: eeye (based on public web pages)
    - ▪ Retina vulnerability scanner is two-part
      - Part-1: Signature-based vulnerability scanner
        - Fast
        - Relatively simple to use
      - Part-2: CHAM... operates like a "hackling-consultant" simulating the methods a hacker would likely use
        - Not fast
        - More difficult to use



Back to Claim-C cont'd

- ❑ Confidence in Claim-C cont'd
  - ❖ Commercial example: eeye cont'd
    - ▪ Retina cont'd
      - CHAM cont'd
        - "Intelligently seeks to compromise target machines" to discover vulnerabilities not found otherwise, including vulnerabilities in custom applications
        - Currently targets HTTP, FTP, SMTP, and POP3 protocols
        - Audit target services for buffer overflows by sending malformed data
        - Newly discovered vulnerabilities in commercial software can be submitted to eeye's vulnerability research team... they will confirm and contact the vendor
    - ▪ Eeye credits use of their automated testing tool in the discovery of announced vulnerabilities
      - Same tool used to discover vulnerabilities in Internet Explorer, Shockwave, MSN Chat, and PNG

Part-3: Summary Observations

❑ 1. MI/AI techniques can and will be used to discover new vulnerabilities faster

❑ 2. The results of #1 can and probably will be used maliciously, increasing the speed at which attacks evolve
- ❖ Widespread acceptance not required, just a few will do
- ❖ Commercial grade tools not required

❑ 3. The results of #1 can be used proactively by organizations to discover vulnerabilities in their software and remediate them before they are exploited
- ❖ Widespread acceptance unlikely
- ❖ Commercial tools required

Summary Observations cont'd

❑ 4. The results of #2 can be countered (mitigated) by
- ❖ Improving our administrative and technical countermeasures
  - ▪ Considering
    - Breadth
    - Depth
    - Agility
  - ▪ See other presentations
- ❖ Developing and purchasing software with less vulnerabilities
  - ▪ See other presentations
- ❖ Using MI/AI technologies to detect and protect ourselves from newly -discovered vulnerabilities
  - ▪ Fighting fire with fire... or more accurately: taking advantage of cheap, abundant machine cycles
  - ▪ The subject of most published research regarding MI/AI for InfoSec

Questions

Discussions

Rebuttal