13th ICCRTS: C2 for Complex Endeavors

Automatically Tracing Information Flow of Vulnerability and Cyber-Attack Information through Text Strings

Topic 6: C2 Assessment Tools and Metrics (also Topics 2 and 7)

Neil C. Rowe, Eric Sjoberg (STUDENT), and Paige Adams (STUDENT)

Naval Postgraduate School

Code CS/Rp, 1411 Cunningham Road, Monterey CA 93943

(831) 656-2462

ncrowe@nps.edu, ejsjober@nps.edu, phadams@nps.edu

Point of contact: Neil Rowe

Abstract

Quick dissemination of information about new vulnerabilities and attacks is essential to time-critical handling of threats in information security, but little systematic tracking has been done of it. We are developing data mining techniques to track the flow of such information by comparing important information-security Web sites, alert messages, and strings in packets to find similar words and sentences. We report on tools we have developed to collect relevant sentences, with a particular focus on comparing sentences from different sources to find patterns of quotation and influence. We report results on some representative pages that indicate some surprising information flows, for which the combination of both word matching and structure matching performed significantly better than either alone. We also report on preliminary work on the front lines of cyber-attack, trying to correlate text in intrusion-detection reports and even attack packets observed on a honeypot with reports of known attacks. These methods could help us automatically locate relevant fixes quickly when being attacked. Our tools will in general enable better design of incident response and incident reporting requirements for organizations, by showing bottlenecks and unused capabilities in the management of vulnerabilities and attacks.

Keywords: Vulnerabilities, alerts, dissemination, World Wide Web, data mining, natural-language processing, cross-document referencing, packets, intrusion-detection systems

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Automatically Tracing Information Flow of Vulnerability and Cyber-Attack Information through Text Strings	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Code CS/Rp,1411 Cunningham Road,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
13th International Command and Control Research and Technology Symposia (ICCRTS 2008), 17-19 Jun 2008, Seattle, WA

14. ABSTRACT

Quick dissemination of information about new vulnerabilities and attacks is essential to time-critical handling of threats in information security, but little systematic tracking has been done of it. We are developing data mining techniques to track the flow of such information by comparing important information-security Web sites, alert messages, and strings in packets to find similar words and sentences. We report on tools we have developed to collect relevant sentences, with a particular focus on comparing sentences from different sources to find patterns of quotation and influence. We report results on some representative pages that indicate some surprising information flows, for which the combination of both word matching and structure matching performed significantly better than either alone. We also report on preliminary work on the front lines of cyber-attack, trying to correlate text in intrusion-detection reports and even attack packets observed on a honeypot with reports of known attacks. These methods could help us automatically locate relevant fixes quickly when being attacked. Our tools will in general enable better design of incident response and incident reporting requirements for organizations, by showing bottlenecks and unused capabilities in the management of vulnerabilities and attacks.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**43**	

1. Introduction

Availability of knowledge of new vulnerabilities and attacks is critical in maintaining security and integrity of computer systems. We have been exploring how such information is transmitted between Internet sites. Studying this is important because it is an essential kind of intelligence gathering for potential hacker and even information-warfare attacks. It is important that channels of information dissemination be efficient in crisis situations, especially for widespread or self-propagating attacks. Without a good understanding of how the channels work, we also cannot improve them very well or develop priority schemes for incident response (Yuill et al, 2000). (Browne et al, 2001) reports that most attacks occur long after security patches are available due to the lack of awareness by system administrators, so better dissemination of attack intelligence should be a high priority.

While some work has been done on "insecurity flow" within software (Moskowitz & Kang, 1997), little systematic attention has been paid to flow of text descriptions of vulnerabilities and attacks concerning software. Proposed languages for reporting vulnerabilities (like OVAL, oval.mitre.org, or that proposed by (Tian et al, 2004)) provide a standardized structure but using them is an imposition on busy system administrators. It would be much more user-friendly to exploit the many bug reports and vulnerability announcements that are already produced in prose. Studies have shown that good dissemination of vulnerability information does not harm systems much, at least as much as dissemination of patches (Arora, Nandkumar, and Telang, 2006).

From our experience we postulate an overall information flow in Figure 1 (Iyer et al, 2003). Attacks are caught by intrusion detection systems, leading system administrators to inspect packets more thoroughly to gain further information. Informal sites like Bugtraq collect reports of system administrators, and broker sites like that for the CVE examine the reports and decide whether to issue a number to the vulnerability and associated (perhaps hypothetical) attack. Further brokers like CERT and other security-practitioner sites then examine the continuing discussion, contact software vendors, and try to provide a definitive statement about the vulnerability and its countermeasures. Note that flow analysis is independent of disclosure issues (Farrow, 2000) since flow can be proprietary or public depending on the kind of information.

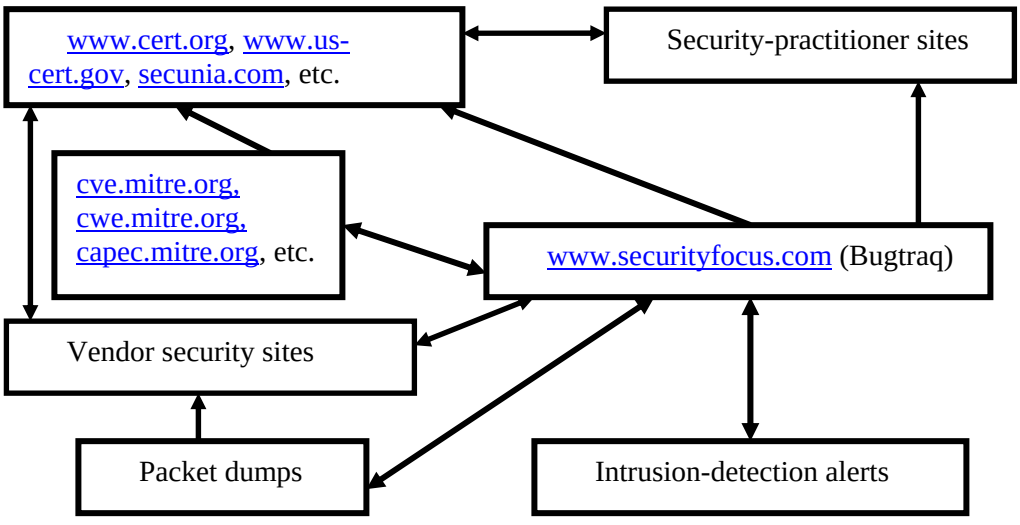


Figure 1: Overall information flow of vulnerability and attack information.

2. Word-matching tools

To examine the current flow of security information, we have developed a number of Java and Python tools to data mine information-security information based on word matching. Earlier prototypes done by students (McVicker, Avellino, & Rowe, 2007) showed promising results from just text keyword matching, so we made that a first step.

2.1 Finding relevant Web sites and matching them

One set of experiments tried to find sentences on the Web mentioning vulnerability information and correlated them to infer who was reading whom. This required our writing first a tool in Java to query Google and Alta Vista and find a set of pages matching a set of vulnerability-associated keywords. In the most extensive experiments reported here, we used the keywords "vulnerability," "ICMP," "packets," "flags," and "footprinting" in an attempt to collect information about ICMP footprinting. We supplemented these links with all outgoing links from those pages whose link text mentioned at least one keyword. Secondly, we wrote a tool to examine the sentences of the found pages to find individual sentences matching the keywords. Only sentences with a weighted match to the keywords exceeding a threshold were returned. The weighting was an estimation of the inverse document frequency: the logarithm of the ratio of the number of occurrences of a word in a sample corpus to the number of occurrences of the given word in the corpus. HTML and other formatting data were removed from sentences before matching. Sentences were carefully delimited using a list of 47 possible sentence-ending patterns. One program analyzed HTML and related formats using a detailed knowledge of HMTL tags; another extracted text from PDF files using the pyPDF utility (cheeseshop.python.org/pypi/pyPdf/1.6). Both used a detailed model of Web URL formats to interpret links and fill in their details. As an example, for the query keywords "ICMP vulnerability flags footprinting", a top-rated sentence at 0.41 was in www.opennet.ru/base/summary/1055867893_2296.txt.html: "An attacker can exploit this vulnerability by sending a simple TCP packet to with the FIN-ACK flags set to a vulnerable machine."

Once sentences were extracted, we compared every pair from different Web sites to find related sentences. (Intrasite copying and citation is common and mostly uninteresting.) This took some time. A threshold of match similarity for output is used based on the weighted sum of the words in common, with the same weighting used previously, but with an extra weight given to connections between sentences of pages having an explicit link between them. We set the threshold to five standard deviations above the mean match score on the extracted sentences, using the mean match score to approximate the mean since the distribution was very close to a Poisson distribution. For example, for the 4692 top pages for the "ICMP flags footprinting" keywords, the mean match score was 0.0488 and the standard deviation was 0.479 (they would be equal for an ideal Poisson distribution). (For the 10,000 top candidates matching 23 keywords associated with insider attacks on computer systems, the mean match score was 0.0400 and the standard deviation was 0.0355.) For the ICMP keywords we used a threshold of 0.25 for match filtering and obtained 35,448 matches. Connections were sorted so the earlier page (judged by its "LastModified" date) came first in the output to enable construction of directed graphs of influence.

We separately tabulated exact matches and inexact matches; the latter were almost always much more common. An example exact sentence match (clearly not a coincidence) rated at 0.604 was between www.unix.org.ua/rfc/bcp0060.html and www.faqs.org/rfcs/bcp/bcp60.html of "Unfortunately, a number of firewalls and load-balancers in the current Internet send a reset in response to a TCP SYN packet that use flags from the Reserved field in the TCP header." An example inexact sentence match rated at 0.54 was between the sentences "Remote attackers could exploit these vulnerabilities to create a denial of service condition, or to execute arbitrary code on

an affected server" and "A remote to unauthenticated attacker could exploit these vulnerabilities to execute arbitrary code or cause a denial of service on an affected system" found on www.juniper.net/security/auto/ vulnerabilities/vuln2558.html and astro.berkeley.edu/~central/ archive/us-cert respectively. In both cases we judged that information flow was from the first page to the second.

We wrote a routine to cluster this data to identify patterns of influence. A threshold argument permits clustering at different levels of detail to see different phenomena. We also summed the ratings for all sentence pairs found between two sites to get a site-pair rating of the degree of information flow. As an example of output, the top-rated inferred flows for the ICMP query were as follows. Some transitivity-like phenomena are apparent, since if site A is similar to site B, and site B is similar to site C, then A is similar to C.

34.966: www.ecst.csuchico.edu to www.yolinux.com
34.943: www.e-infomax.com to www.ecst.csuchico.edu
34.943: www.ecst.csuchico.edu to www.uni-kiel.de
34.943: www.linuxdig.com to www.ecst.csuchico.edu
31.713: docs.mandragor.org to www.ecst.csuchico.edu
31.696: www.ecst.csuchico.edu to www.arameya.com
28.713: www.e-infomax.com to www.uni-kiel.de
28.713: www.linuxdig.com to www.e-infomax.com
28.713: www.linuxdig.com to www.uni-kiel.de
28.710: www.e-infomax.com to www.yolinux.com
28.710: www.linuxdig.com to www.yolinux.com
28.710: www.uni-kiel.de to www.yolinux.com
28.426: www.cs.wisc.edu to www.ecst.csuchico.edu
27.397: docs.mandragor.org to www.arameya.com
26.796: docs.mandragor.org to www.e-infomax.com
26.796: docs.mandragor.org to www.linuxdig.com
26.796: docs.mandragor.org to www.uni-kiel.de
26.793: docs.mandragor.org to www.yolinux.com
26.793: www.e-infomax.com to www.arameya.com
26.793: www.linuxdig.com to www.arameya.com
26.793: www.uni-kiel.de to www.arameya.com
26.788: www.arameya.com to www.yolinux.com
24.035: docs.mandragor.org to www.cs.wisc.edu
24.032: www.cs.wisc.edu to www.arameya.com
23.792: www.armware.dk to www.faqs.org
23.783: www.faqs.org to ietfreport.isoc.org
23.783: www.unix.org.ua to www.faqs.org
23.755: www.ietf.org to www.faqs.org

2.2 Classification of sentence similarities

Two sentences on two pages may be similar for several reasons. Exact matches are usually beyond the limits of coincidence. Exact matches could be:

- Normal routine copying of pages, as when a Web site collects important papers on security. The "TCP SYN" exact match given above can be inferred to be an example of the first because the page names are also similar, bcp0060.html and bcp60.html. Pages with numerous high-similarity sentences strongly support this hypothesis.
- Common authorship on sites. We excluded matching of sentences within the same site (the same domain), but businesses often buy multiple site names to appeal to different

audiences. An example is an exact match between www.demboo.info/Carbon-cheats.htm and www.mulax.info/Sims-cheats.htm of the sentence "For walkthroughs, cheats and tips call 09067 53 54 55 This is a fully automated system that provides gameplay hints and playing tips for most of the games in the Electronic Arts range." A high site-pair rating of similar sentences supports this hypothesis, but it can be distinguished from routine copying by having significantly different file names between those of similar sentences.

- Acknowledged citation, particularly if the text is quoted or indented. This rarely occurred in our security pages, but occurs much more with traditional journalism. This can be distinguished by introductory words in a previous sentence such as "says", "explains", "stated", "according to", "further information", etc.
- Plagiarism, not as uncommon as one would hope. Of course, security crises can require getting accurate information out quickly, and copying someone's words without citation may occur. We did not see any obvious examples in our test cases, but they would be hard to prove.
- "Boilerplate", formalized statements to accomplish some legal or policy objective. For instance, www.securiteam.com/securitynews/5RP0E204UA.html has "Additional Information: For the most up-to-date information regarding these vulnerabilities, please visit the CERT/CC Vulnerability Notes Database at: http://www.kb.cert.org/vuls/" and "Please note that the test results summarized above should not be interpreted as a statement of overall software quality.", and both sentences also occur in astro.berkeley.edu/~central/archive/us-cert. Boilerplate can be inferred from the use of particular words such as "information", "please visit", "please note", and "should not be interpreted".

Inexact matches between sentences on different pages could be:

- Common authorship on sites. For instance, www.konde.info/Nothing-cheats.htm says "IGN is the ultimate Spider-Man: The Movie resource for trailers, screenshots, cheats , walkthroughs ... " and www.mulax.info/Games-cheats.htm says "IGN PS2 is the ultimate resource for PlayStation 2 trailers, screenshots, cheats, walkthroughs ... ". It is unlikely that anyone other than the same author would have strung those four particular nouns in succession at the end.
- Acknowledged citation. For instance, cert.pol34.pl/news/annall.htm has the sentence "According to Microsoft Advisory (935423), in order for this attack to be carried out, a user must either visit a Web site that contains a Web page that is used to exploit the vulnerability or view a specially crafted e-mail message or email attachment sent to them by an attacker." This is usually signaled by introductory words in the sentence.
- Acknowledged paraphrase. For example, cert.pol34.pl/news/annall.htm says "According to the US-CERT there is publicly available exploit code for multiple vulnerabilities in Sun Java Runtime Environment (JRE)" and astro.berkeley.edu/~central/archive/us-cert says "Publicly available exploit code exists for this vulnerability, and US-CERT has monitored incident reports that indicate that this vulnerability is being actively exploited."
- Unacknowledged paraphrase. We see much that looks like this, though is hard to prove. It often occurs in attempts to translate more technical language into more accessible language.
- Boilerplate, such as required legal notices. For instance, many pages at astro.berkeley.edu/~central/archive/us-cert begin with "Further information is available in the following US-CERT Vulnerability Note".
- Accidental similarities. These can occur with commonly repeated language. For instance, www.freerepublic.com/focus/keyword?k=msie says "An attacker could use a specially crafted web page to exploit the vulnerability and take control of a system, warned Danish security firm Secunia" and astro.berkeley.edu/~central/archive/us-cert says "An attacker could exploit these vulnerabilities by using specially crafted network traffic, by

convincing you to click on a specially crafted URL, or by convincing you to open a specially crafted Office document". But it is unlikely that the two sentences are referring to the same vulnerability, since they are both describing Web client vulnerabilities in a general way; previous sentences on their pages distinguish their motivating subjects more precisely.

3. Structure-based sentence matching

To explore security-assertion matching in more detail, we conducted additional experiments with the pages and sentences found in our first experiments. Many authors have distinctive styles of sentence structure. For example, one author may prefer multiple adjectives within a sentence while another may prefer prepositional phrases. The occurrence and frequency of adjectival and prepositional phrases can define an author's writing style. This idea is currently being used to detect pseudonymous writing (Rao & Rohtagi, 2000). When authors of vulnerability and attack information copy or use preexisting documents as templates, they preserve the characteristic style of the original author, and this may be detectable.

3.1 Mining methodology

For these experiments we used the Python programming language because of its ease and flexibility in working with text strings, the free availability of several useful modules for performing web-mining functions, and our experience with the Python-based Natural Language Toolkit (NLTK) for NLP routines (Sourceforge, 2007). Useful in particular were the `mechanize` and `BeautifulSoup` modules. The `mechanize` module provides a means for programmatic web browsing and allowed us to retrieve and iterate over Web pages. The `BeautifulSoup` module is a HTML/XML parser that can compensate for invalid markup structure and provides navigation, search, and modification functions for the parse tree. We used it to extract specific data elements from a parsed forum page.

In these experiments we focused on extracting data from two sources: vulnerability note entries from Carnegie-Mellon's CERT database, and posts from the Bugtraq computer security forum operated on the SecurityFocus website owned by Symantec Corporation. As an archive of a high-volume mailing list, Bugtraq's forum contains early notification and discussion of new security vulnerabilities, while the CERT database contains descriptions of vulnerabilities that have been formally verified and written up by the CERT. Accordingly, we expected that a new vulnerability would first be reported in the Bugtraq forum, and then, once verified, appear in the CERT database. We wanted to trace the information vector from initial vulnerability indication to verification and resolution.

3.2 Comparisons

The data retrieved from the CERT and Bugtraq entries had few undocumented relationships. There were approximately 2,500 entries in the CERT database and 25,750 in the Bugtraq forum that matched seven test topics we selected. Based on these results, we elected to build a test set for our comparison algorithms on a larger previously collected data set that contained sentences resulting from a search on insider security attacks. In this data set were 7,085 sentence pairs, which were read and manually categorized as being related or unrelated.

Our methodology was to use low-level natural-language processing to provide information about the part-of-speech sequence in the sentences (nouns, verbs, articles, adjectives, etc. and their ordering). We utilized an N-Gram tagger provided by the NLTK to determine the parts of speech of sentence tokens (typically words) (Jurafsky & Martin, 2000). It tags a token based on that token and its N-1 predecessor tokens. It assigns a tag by looking up the most likely tag for the sequence of tokens and tags based on the data on which it was trained. We trained trigram, bigram, and unigram taggers using the Wall Street Journal data subset of the Penn Treebank. The

trigram tagger was the primary tagger since it provided the most context for tagging individual tokens. The trigram tagger backed off (that is, resorted) to first the bigram and then the unigram tagger when there was insufficient data for a three-token sequence. The unigram tagger backed off to a regular expression tagger and then a default tagger. The regular expression tagger attempted to provide correct tags for regularly occurring tokens such as dates, gerunds, simple past tense verbs, and URLs.

To compare two sentences, several algorithms were implemented and compared (see Figure 2). We used the f-score (the geometric mean of recall and precision) as our performance metric. The first method ("Keywords") calculated only the ratio of common keywords in both sentences. The second method ("TS") calculated the ratio of common tags and tag bigrams in both sentences. The third method ("TTS") multiplied the number from the second method with the ratio of tokens in common between the two sentences. For this method, tokens that were tagged as articles, conjunctions, and pronouns were ignored since they occur fairly often. For the final method ("TTS-A"), the ratio of similar tags, tokens, and bigram tags were averaged together. This provided a more comprehensive metric since it included the similarity weighing the words in the sentence, the type of words in the sentence, and the structure of the sentence (captured by the tag bigrams) equally.

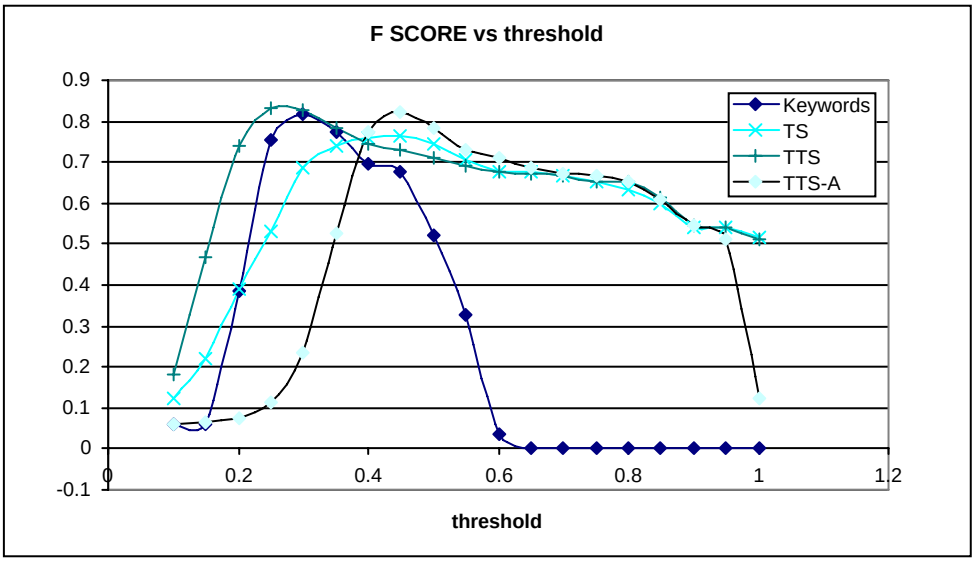


Figure 2: Relative performance of four sentence-comparison algorithms.

3.3 Results

Overall, the tagger appeared quite successful in detecting similarity between two sentences. Since many security-related websites use well-formed English to describe the potential bug or security vulnerability, the Wall Street Journal data proved adequate for tagging these sentences. Where regularly formed English was not found, the unigram tagger provided the most likely tag given that word.

Figure 2 shows our final algorithm TTS-A achieved its highest f-score of .83, when precision was .96 and recall .73. Figure 3 shows more details of the tradeoff between recall and precision. Compared to keyword matching alone, this is a 37% increase in recall for the same level of precision. Surprisingly, our algorithm achieved over 99% precision with 50% recall. This

indicates that just looking at the tokens, their tags, and a partial ordering of the tags suffices to establish a correlation.

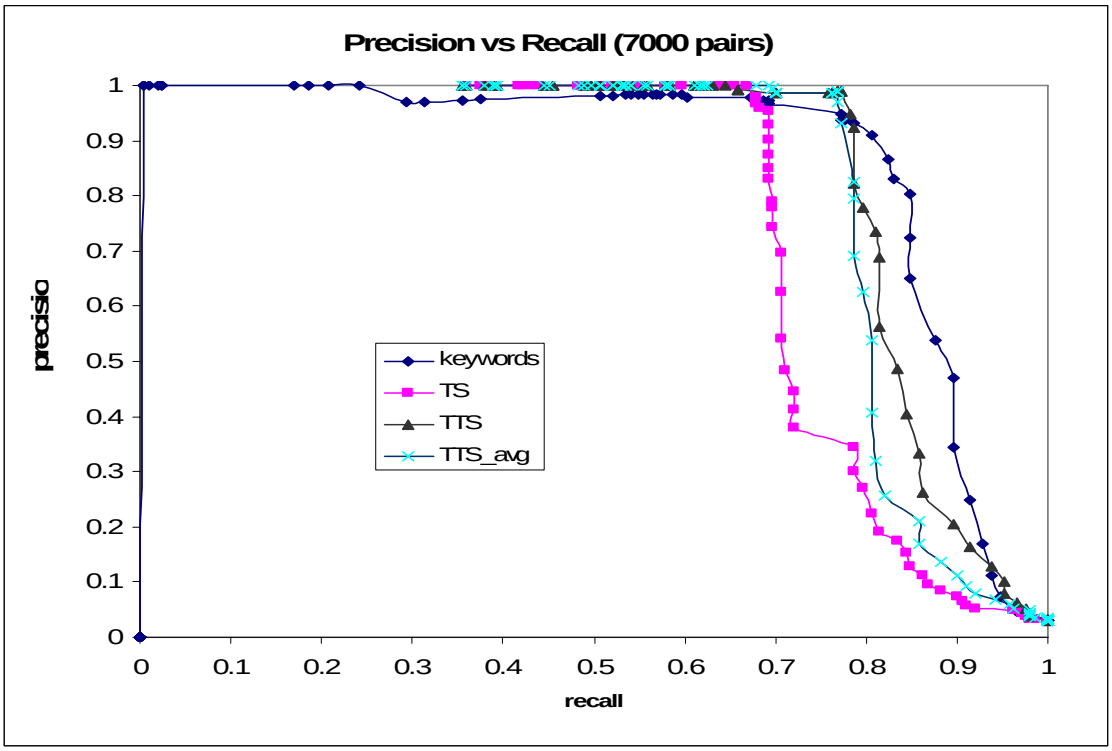


Figure 3: Recall-precision curve for the 7000 test sentence pairs.

The sentences for which comparison of tokens, tags, and structure correctly identified the match or lack thereof while keywords alone did not largely appear to be those in which a particular substring was common to both sentences. An example is the pair with the accidental substring "that bundles together many different malware tools":

- "MPack is a powerful kit that bundles together many different malware tools."
- "The kit is a professionally developed collection of back-end web components built on PHP that bundles together many different malware tools."

Structure matching permits matching to be more tolerant to parsing errors. For example in the below pair, the comprehensive matching correctly predicts a relationship between the two sets of text even though the parsing algorithm failed to properly parse the text into the correct set of sentences.

- "It monitors global risks and threats including global warming, terrorism, cybercrime, economic espionage, etc. It champions security at home and at work."
- "It monitors global risks and threats including global warming, terrorism, national disasters and health emergencies, cybercrime, economic espionage, etc. It also analyses issues and trends in the struggle for geopolitical hegemony, the pursuit of energy security and environmental security, the cultivation of human rights, and the strengthening of democratic institutions."

Another interesting result is that with a threshold of about 0.45, the token, tag, and structure method achieves a 1.0 for precision and approximately 0.66 for recall on our test set. Any pair of sentences achieving a score above 0.7 means that one sentence is a direct copy of the other with some minor errors in parsing or punctuation.

Figures 4 and 5 show histograms of all the scores assigned to the 7000 sentence pairs. 95% of the pairs received scores below 0.3, indicating a low probability of a match. The peak in scores between 0 and 0.2 represents random coincidences of nouns, articles, and verbs.

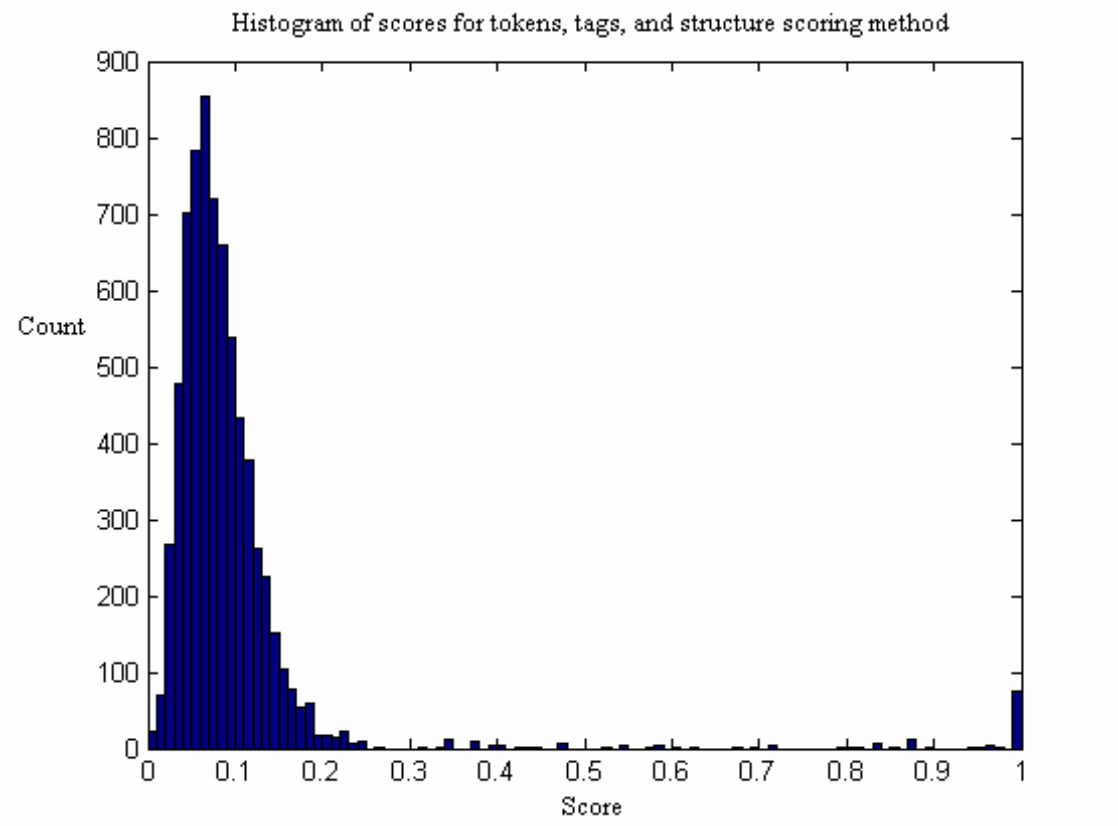


Figure 4: Histogram of scores for combined matching of tokens, tags, and structure.

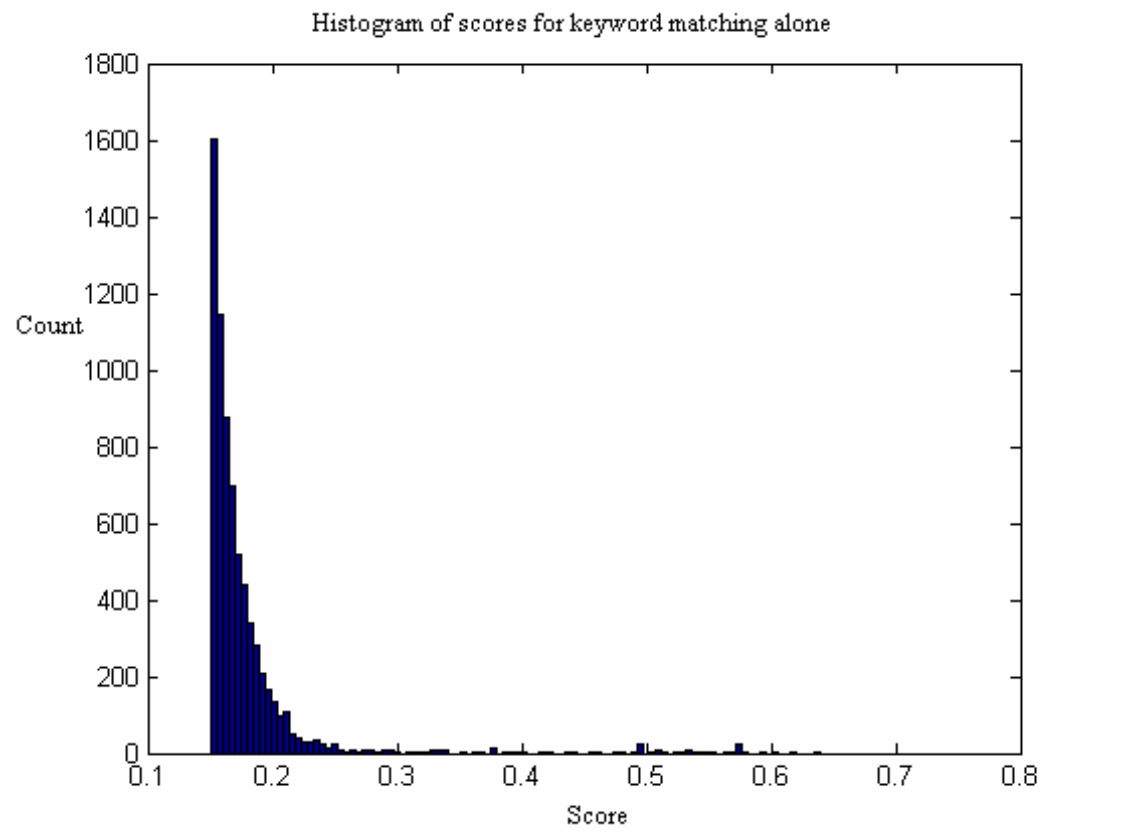


Figure 5: Histogram of scores for keyword matching alone.

It was difficult to confirm the relationship of many pairs receiving scores between 0.3 and 0.45. For example, the sentences below which scored 0.44 discuss a Zogby poll on the same subject. Particularly intriguing is that these sentences are using the same poll and the number is different, whether accidentally or deliberately.

- "A Zogby poll of New Yorkers' opinions about the 9/11 investigation, released last month, indicated that 49 percent of New York City residents and 41 percent of New York state residents believed that some federal officials 'knew in advance that attacks were planned on or around September 11, 2001, and that they consciously failed to act.'"
- "Consider another Zogby poll from August 2004, which found that 63 percent of New Yorkers under 30 believe some U.S. leaders 'knew in advance that attacks were planned on or around September 11, 2001, and that they consciously failed to act.'"

Of particular interest were sentence pairs where all algorithms failed to see that the sentences were related (Figure 6). Many of these sentences appear to be completely different in structure even though they discuss identical subjects. Higher-level processing with full parser may be necessary to recognize such pairs. Particularly helpful might be to focus on noun-noun references as this has been very helpful in parsing captions (Guglielmo & Rowe, 1996). The Appendix gives more examples of specific matches found by our programs and their ratings.

Sentence 1	Sentence 2
"One strain of scam email makes the bogus claim that recipients have won one of the much sought after devices in a bid to trick prospective marks into visiting a malware loaded site."	"Email recipients are sent a bogus email informing them that they have won a new iPhone, in reality the email contained malware designed to subvert and compromise the user's computer."
"NanoScan is a rapid, light scanner that currently detects over 750,000 active viruses, spyware, Trojans and other malware within just one minute."	"Panda Software has launched the mini, customisable version of NanoScan, the instant virus scanner from Panda Software, designed to detect active malware on a PC in less than one minute."
"According to a 2005 FBI Cyber Crime Study, 90 percent of small businesses had at least one cyber security incident within the past year."	"In fact, of the 500 companies that responded to a recent FBI survey, 90 percent said they'd had a computer security breach, and 80 percent of those said they'd suffered financial loss as a result."
"Using ideological attraction, the Soviets successfully recruited many high-level spies."	"At that time period the Soviets recruited their spies using ideological motivation."

Figure 6: Sentence pairs where our structure-matching methods fail.

4. Correlating attack reports with packets

A final step in using descriptive information about attacks is in correlating it to the details of observed attacks. If we can automate the latter as well as the indexing of resource sites such as those of Bugtraq and the CERTs, we could recognize attacks automatically within a second after they occur in a more general way than that provided by intrusion-detection systems. This would be helpful since, as (Lai & Hsia, 2007) points out, many administrators are too busy to do anything about their vulnerabilities until they receive attack reports.

A good way to accomplish this last step is to continuously collect attack data on a honeypot, a computer deliberately intended for no purpose other than to be attacked. We have done some first steps in exploring this using a honeypot we have been running to study deception methods (Rowe and Goh, 2007). Honeypots provide plenty of data about common untargeted attack methods on the Internet, its "background radiation" (Pang et al, 2004).

4.1 Intrusion alert records

Two useful kinds of data obtainable from a machine under attack are the record of suspicious events and the full packet records. An intrusion-detection system can provide the first kind for known attacks. For instance, one record from the Snort intrusion-detection system running on our honeypot was:

Date: 2007-09-12 Time: 15:46:56.148-07 Alert_code: 1394 Alert_description: SHELLCODE x86 NOOP IP_address_1: 89.26.217.22 Port_#_1: 4310 IP_address_2: 192.168.0.3 Port_#_2: 445 time_to_live: 118

These alerts are triggered by Snort production rules of a relatively simple syntax that are created by programmers who study attack traffic. Although there are periodic updates, it may take a while for a rule for a new attack to get implemented. However, there are a sufficient number of general rules that Snort and other intrusion-detection system can trigger to recognize at least something in a new attack, because attacks often reuse parts of others.

Snort alert codes are indexed with some reference information, sometimes to CERT or Bugtraq sites. These can be looked up at the Snort site and correlated with the information in their referents. For instance, rule 1394 that triggered the above alert has the following description at www.snort.org, providing a good number of useful keywords for further lookup.

GEN:SID 1:1394
Message SHELLCODE x86 NOOP
Summary This event is generated when an attempt is made to possibly overflow a buffer. The NOOP warning occurs when a series of NOOP (no operation) are found in a stream. Most buffer overflow exploits typically use NOOPs sleds to pad the code.
Impact This might indicate someone is trying to use a buffer overflow exploit. Full compromise of system is possible if the exploit is successful.
Detailed Information This rule detects a large number of consecutive NOOP instructions used in padding code. It's not specific to a particular service exploit, but rather used to try and detect buffer overflows in general. It is common for buffer overflow code to contain a large sequence of NOOP instructions as it increases the odds of successful execution of the useful shellcode.
Affected Systems Any x86 programs.
Attack Scenarios An attacker uses a buffer overflow exploit which contains the following payload: 90 90 90 90 90 90 90 90 90 90 /bin/sh
Ease of Attack Simple.
False Positives High, This event may be generated by applications such as ftp and http when binary data is being transferred. A false Positive can be generated if the snort sensor detects text from an IRC client or any other application that passes data plaintext. The event is generated if snort detects several (a) characters in a row - such as 'aaaaaaaaaa'.

4.2 Packet analysis

Since new attacks and even many well-known attacks may not trigger an alert system, we may need to find them in the background traffic. Text strings occur surprisingly often in attack traffic, and other strings can be constructed from nontext data, though the text strings tend to be more fruitful since they come prechunked. Two examples of packet dumps from TCPDump on our honeypot were:

```
09/14-00:47:21.626361 131.120.18.41:53 -> 192.168.0.3:3559 UDP
TTL:111 TOS:0x0 ID:15349 IpLen:20 DgmLen:145 Len: 117
47 59 81 83 00 01 00 00 00 01 00 00 02 67 63 06  GY...........gc.
5F 6D 73 64 63 73 08 55 53 4E 42 41 52 4F 4E 05  _msdcs.USNBARON.
6C 6F 63 61 6C 00 00 06 00 01 00 00 06 00 01 00  local...........
00 00 00 00 40 01 41 0C 52 4F 4F 54 2D 53 45 52  ....@.A.ROOT-SER
56 45 52 53 03 4E 45 54 00 05 4E 53 54 4C 44 0C  VERS.NET..NSTLD.
56 45 52 49 53 49 47 4E 2D 47 52 53 03 43 4F 4D  VERISIGN-GRS.COM
00 77 A1 C8 65 00 00 07 08 00 00 03 84 00 09 3A  .w..e..........:
80 00 01 51 80                                   ...Q.

09/16-22:43:13.038582 131.120.18.41:53 -> 192.168.0.4:1052
UDP TTL:111 TOS:0x0 ID:13512 IpLen:20 DgmLen:137 Len: 109
FF FA 81 83 00 01 00 00 00 01 00 00 08 64 6F 77  .............dow
6E 6C 6F 57 64 0D 77 69 6E 64 6F 77 73 75 70 64  nloWd.windowsupd
61 74 65 03 63 6F 6D 00 00 01 00 01 C0 15 00 06  ate.com.........
00 01 00 00 0E 10 00 35 03 6E 73 31 04 6D 73 66  .......5.ns1.msf
74 03 6E 65 74 00 06 6D 73 6E 68 73 74 09 6D 69  t.net..msnhst.mi
63 72 6F 73 6F 66 74 C0 23 77 A1 A5 3D 00 00 03  crosoft.#w..=...
84 00 00 02 58 00 09 27 C0 00 00 03 84           ....X..'.....
```

Here the first two lines give the packet header information, the left side below gives the raw bytes in hexadecimal, and the right side translates it into alphanumeric characters if possible. The first packet refers to a low-security site USNBARON that has been used for attacks, and the second refers to downloWd, a favorite hacker spelling. But not all the strings are interesting; the first packet also refers to nstld.verifsign-grs.com, a standard address for checking certificates, and the second also refers to windowsupdate.com, the standard Windows updating site.

We can connect these strings to attack intelligence by supplying them to a Web search engine. In general, we can look up any substantial character strings we find in a packet, including also the individual words like "root-servers", "Verisign", and "msnht" above, in our databases of text about attacks created using the methods of sections 2 and 3 above. Techniques can be similar to those of forensics on malicious code which also exploit hidden strings in the code. It does not matter whether the strings are functional in the attack or not – sometimes they are artifacts of the attacker software, and sometimes they are just bragging – because they can all provide identification and classification clues. Even subtle word clues can help, like "MARB" and "MEOW"s in the following portion of a packet we received. Sure enough, doing a Google lookup on those two words finds a reference (Parker, 2004) which explains their significance.

```
05 00 00 03 10 00 00 00 A8 06 00 00 E5 00 00 00  ................
90 06 00 00 01 00 04 00 05 00 06 00 01 00 00 00  ................
00 00 00 00 32 24 58 FD CC 45 64 49 B0 70 DD AE  ....2$X..EdI.p..
74 2C 96 D2 60 5E 0D 00 01 00 00 00 00 00 00 00  t,..`^..........
70 5E 0D 00 02 00 00 00 7C 5E 0D 00 00 00 00 00  p^......|^......
10 00 00 00 80 96 F1 F1 2A 4D CE 11 A6 6A 00 20  ........*M...j. 
AF 6E 72 F4 0C 00 00 00 4D 41 52 42 01 00 00 00  .nr.....MARB....
00 00 00 00 0D F0 AD BA 00 00 00 00 A8 F4 0B 00  ................
20 06 00 00 20 06 00 00 4D 45 4F 57 04 00 00 00   ... ...MEOW....
A2 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  ...............F
38 03 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  8..............F
00 00 00 00 F0 05 00 00 E8 05 00 00 00 00 00 00  ................
01 10 08 00 CC CC CC CC C8 00 00 00 4D 45 4F 57  ............MEOW
E8 05 00 00 D8 00 00 00 00 00 00 00 02 00 00 00  ................
```

The headers of attack packets also provide useful indexing words to the attack, particularly the packet-length ("Length"), time-to-live ("TTL"), type-of-service ("TOS"), and protocol-name (at the end of the first line) fields. To exploit these, we give the field name and the associated value as keywords. IP addresses are useful since they can be looked up in registry sites like the ARIN registry (www.arin.net) for information about what they are. More specific classification of the type of usage (such as the specific service using HTTP) can be obtained by simple methods of approximate assembly of the packets, and this helps classify the attack. Subtle clues in the packets can also be learned by using techniques such as support-vector machines (Li, Wang, & Luo, 2006). Further packet partitioning can be done using techniques from digital forensics in extracting hidden files from disks (Garfinkel, 2007).

4.3 Keyword lookup

Once we have accumulated keywords from the attack description (with intrusion detection) or the packet itself, we can find out what vulnerabilities and attacks are associated with them. Many words used in attack packets are common generic words, so it is important to eliminate them. (In addition, we saw some true text messages on the honeypot, such as those from UDP phishing.) We can keep a list of such ignorable words much like stopword lists used in conventional text data mining. Currently we use one of 638 words (mostly from our previous data-mining work but with some additions specific to packets), including prepositions, adverbs, adjectives, pronouns, and overly-general nouns and verbs, as well as names of common Internet sites. Example entries are “into”, “rather”, “entire”, “ourselves”, “rather”, “order”, “fact”, “mr”, “root”, “windows”, “need”, “please”, “2007”, “Adobe”, and “Microsoft”.

As an experiment, we took 1,632,374 lines of TCPDump output from five days of our honeypot run in September 2005, which was about 14 megabytes of packet data. We grouped together all consecutive packets with the same IP addresses (source and destination) and port numbers, and extracted their text strings as sets, one per packet sequence. After eliminating stopwords, we obtained only 406 unique string sets averaging 5.7 words per set, each characterizing an attack type or variant. An example is "aof baronus gss ntlmssp usn". Many pairs of these sets had substantial overlap, like that between the previous example and "aof baronus gss mww ntlmssp usn vge yeo". The strings critical to attacks are those common to pairs. A simple algorithm to get the critical strings is to successively find pairs of sets which overlap in all but one string (corresponding to attack instances in which only one string differed). Applying this to the 406 sets from the experiment gave us 67 additional strings of significantly higher precision.

Our remaining words can then be looked up in an index of vulnerability and attack information using the methods of section 2. For this lookup it is usually desirable to search for references that match the conjunction of terms rather than the disjunction since most attacks depend on a specific conjunction of features in the packet to be effective, and precision tends to be low for disjunctive queries with these kinds of words. However, packets often form sequences based on their originating IP address and ports. Collecting all the strings for the packet sequences gives us large numbers of keywords for which matches on only a significantly large subset should be sufficient. To avoid missing useful data, we should do lookups on both the individual packets and packet sequences.

What if we fail to find any reasonably specific vulnerability or attack information on a particular attack packet sequence? That can actually be good because we may be seeing a new attack. We should be particularly interested if unusual strings (as judged by frequency counts) occur in the packet strings, as these may be our first warning of new attack methods. We should record the sets and, if possible, post them on bulletin boards to seek additional information.

5. Future Work

We have developed a variety of tools for data mining of vulnerability and attack information. The next step is to upscale our tools to collect a much larger chunk of the current information flow. This will require some database design and a bank of computers running continuously to fill the database.

The tools themselves need to be tuned. With part-of-speech tagging, we could achieve better results than with keyword matching alone, and the combination of the two worked even better. We estimate that using even more sophisticated natural-language processing techniques might improve the results and discover relational patterns not evident with our current techniques. In particular, semantic analysis (looking at the "meaning" of the text) would provide the opportunity to discover related textual information even if a sentence were rewritten using a different word structure and vocabulary. Some recent work has shown good results in detecting semantic entailment in two separate pieces of text.

Information flow analysis of vulnerability and attack information can be used in a variety of dissemination architectures. It could support centralized reporting and dissemination, or it could support a peer-to-peer sharing of new information (Baquero & Lopes, 2003), depending on the resources and philosophy of the users. Identification of new kinds of attacks in a protocol could at least prompt system administrators to disable service of the protocol on their routers and other networking equipment, a simple approach that can immediately reduce attack damage significantly (Lai & Hsia, 2007). It could supply data for accurate modeling of vulnerability reporting (Browne et al, 2001). It could also provide the necessary data for design of good randomization strategies for software based on what vulnerabilities tend to be exploited (Iyer et al, 2003), or deception strategies (Rowe & Goh, 2007). Detection of new attack variants in packets could be worth money to its discovers since some organizations pay for new exploits (Kannan, Telang, & Xu, 2004).

6. Conclusions

We have developed parts of a new approach to real-time information security, an approach that automatically recognizes text associated with attacks and correlates them with Web information about them. This could provide useful "information flow" analysis of how attack and vulnerability intelligence is created and disseminated, indicating possible bottlenecks and redundancies. It could also provide a basis for automatic real-time defense against attacks even if not much is yet known about them. But much work needs to be done to build the database of significant size that will be necessary, and set up automatic updates to it.

References

Arora, A., Nandkumar, A., & Telang, R. 2006. Does information security attack frequency increase with vulnerability disclosure? An empirical analysis. *Information Systems Frontier*, Vol. 8, No. 5, 350-362.

Baquero, C., & Lopes, N. 2003, October. Towards peer-to-peer content indexing. *ACM SIGOPS Operating Systems Review*, Vol. 37, No. 4, pp. 90-96.

Browne, H., Arbaugh, W., McHugh, J., & Fithen, W. 2001, May. A trend analysis of exploitations. *Proc. IEEE Symposium on Security and Privacy*, pp. 214-229.

Farrow, R. 2000. Vulnerability disclosure debate. Network Magazine, October 2000. www.spirit.com/Network/net0800.html, Retrieved December 12, 2007.

Garfinkel, S. 2007. Carving contiguous and fragmented files with fast object validation. *Digital Investigation*, 45, 2-12.

Guglielmo, E., and Rowe, N. 1996, May. Natural language retrieval of images based on descriptive captions. *ACM Transactions on Information Systems, 14*, 3, 237-267.

Iyer, R., Chen, S., Xu, J., & Kalbarczyk, Z. 2003, October. Security vulnerabilities – from data analysis to protection mechanisms. *Proc. 9th Intl. Workshop on Object-Oriented Real-Time Dependable Systems*, pp. 331-338.

Jurafsky, D., and J. Martin. 2000. *Speech and language processing*. Prentice Hall.

Kannan, K., Telang, R., and Xu, H. 2004, January. Economic analysis of the market for software vulnerability disclosure. *Proc. 37th Hawaii Intl. Conf. on System Sciences*, p. 8.

Lai, Y.-P., & Hsia, P.-L. 2007, June. Using the vulnerability information of computer systems to improve the network security. *Computer Communications*, Vol. 30, No. 9, 2032-2047.

Li, B., Wang, Q., & Luo, J. 2006, December. Forensic analysis of document fragment based on SVM. *Proc. Intl. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Pasadena, CA, 236-239.

McVicker, M., P. Avellino, and N. Rowe. 2007. Automated retrieval of security statistics from the World Wide Web. *Proc. 2007 IEEE Workshop on Information Assurance*, United States Military Academy, West Point, New York, pp. 355–356.

Moskowitz, I., & Kang, M. 1997. An insecurity flow model. *Proc. 1997 Workshop on New Security Paradigms*, Langdale, UK, pp. 61-74.

Pang, R., V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. 2004. Characteristics of Internet background radiation. *Proc. 4th ACM SIGCOMM Conference on Internet Measurement*, Taormina, IT, pp. 27-40.

Parker, D. 2004. Examining a public exploit, part II. Retrieved from www.securityfocus.com/infocus/1801, September 15.

Rao, J., and P. Rohatgi. 2000. Can pseudonymity really guarantee privacy? *Proc. of the 9th USENIX Security Symposium*.

Rowe, N., and H. Goh. 2007. Thwarting cyber-attack reconnaissance with inconsistency and deception. *Proc. 8th IEEE Information Assurance Workshop*, West Point, NY, June, pp. 151-158.

Sourceforge. 2007. *Natural Language Toolkit.* Retrieved from http://nltk.sourceforge.net, September 1.

Tian, H., Huang, L., Zhou, Z., & Luo, Y. 2004, May. Arm up administrators: automated vulnerability management. *Proc. 7th Intl. Symposium on Parallel Architectures, Algorithms, and Networks*, pp. 587-593.

Yuill, J., Wu, F., Settle, F., Gong, R., Forno, R., Huang, M., & Asbery, J. 2000, October. Intrusion-detection for incident-response, using a military battlefield-intelligence process. *Computer Networks*, Vol. 34, No. 2, 671-691.

Appendix. Example sentence matches

We show here some more examples (both successful and unsuccessful) of the sentence matching. We give in order the dates of the pages, the text of the pages, the URLs of the pages, and the rating we computed using both keywords and structure matching.

Wed Dec 31 16:00:00 PST 1969
Mon Nov 22 08:12:44 PST 2004
Symantec (2003) ?Symantec Internet Security Threat Report Sees Increase in Blended Threats, Vulnerabilities and Internet Attacks

Press Release,?Symantec (2003) ?Symantec Internet Security Threat Report Sees Increase in Blended Threats,Vulnerabilities and Internet Attacks?
http://72.14.253.104/search?q=cache:BSGuNyNGVdIJ:www.nvpcug.org/Newsletter/7July05.pdf
http://secureflorida.org/clientuploads/C-SAFE/CSAFEcybersecuritymanual.pdf
0.91751

Mon Jul 02 09:26:18 PDT 2007
Tue Jul 03 15:10:02 PDT 2007
Lack of Mac malware baffles experts - vnunet.com Apple's Mac OS X remains almost completely free of any sort of malware threat despite several years of availability, a significant market share, and even an entire month dedicated to pointing out its flaws.
Apple's Mac OS X remains almost completely free of any sort of malware threat despite several years of availability, a significant market share, and even an entire month dedicated to pointing out its flaws.
http://www.b12partners.net/mt/archives/macintosh/
http://www.vnunet.com/vnunet/news/2186013/dearth-mac-malware-continues
0.91394

Mon Aug 20 13:57:51 PDT 2001
Tue Jul 03 04:09:15 PDT 2007
A debate has raged for some time over whether the major threat to system security arises from attacks by "insiders" or by "outsiders." Insiders have been blamed for causing 70 to 80 percent of the incidents and most of the damage (Lewis, 1998).
Insiders have been blamed for causing 70 to 80 percent of the incidents and most of the damage (Lewis, 1998).
http://www.aci.net/kalliste/tic.htm
http://web.elastic.org/%7efche/mirrors/www.jya.com/tic.htm
0.70866

Wed Dec 31 16:00:00 PST 1969
Tue May 31 00:21:30 PDT 2005
Botnets or zombie networks are groups of computers that have been infected by malware that allow the malware to control the infected PC and use it to send spam or launch distributed denial of service (DDoS) attacks.
Zombie networks are groups of computers that have been infected by malware that allows the author to control the infected PC and use it to send spam or launch DDoS attacks.
http://weblog.infoworld.com/techwatch/archives/cat_security.html
http://www.attrition.org/pipermail/isn/2005-May.txt
0.68167

Wed Dec 31 16:00:00 PST 1969
Wed Dec 31 16:00:00 PST 1969
Adware Installation Trick 3: Outright Lying How it works: malware may even be labeled as something else entirely, such as a well-known piece of software or a crucial component of the computer operating system.
Adware Installation Trick 1: Piggybacking How it works: malware may come bundled with a legitimate piece of software the user actually wants, such as a game or emoticon.
http://www.greatarticleshere.com/aid32813/Adware-How-to-Beat-the-Sneakiest-Software.html
http://www.simplyyourarticles.com/aid16923/Adware-How-to-Beat-the-Sneakiest-Software.html
0.58637

Wed Dec 31 16:00:00 PST 1969
Fri Aug 12 18:42:55 PDT 2005
In the first quarter of 2007, security firm Sophos PLC identified 23,864 new malware threats, more than double the 9,450 the company found in the same period last year.
The firm reported last week that it had detected 7,944 new pieces of such malware in the first six months of this year ? almost 60 percent more than the same time last year.
http://203.29.124.140/feed/single/21
http://www.cwalsh.org/isnd/archives/2005_07.html
0.46503

Thu Mar 27 02:20:40 PST 2003
Tue May 31 00:21:30 PDT 2005
In the paper statistics can be found on computer crime vulnerabilities, computer crime incidents, computer security incidents, malicious attacks, etc. which could include crimes, attempts at crimes, etc. but probably also non-criminal conduct.
These fix some vulnerabilities, which can be exploited by malicious people to conduct cross-site scripting attacks, bypass certain security restrictions, gain knowledge of potentially sensitive information and compromise a user's system.
http://www.ulapland.fi/home/oiffi/enlist/resources/HeuniWeb.htm
http://www.attrition.org/pipermail/isn/2005-May.txt
0.42534

Wed Dec 31 16:00:00 PST 1969
Wed Dec 31 16:00:00 PST 1969
Botnet computers are machines (generally running one of the notoriously insecure Windows OSes) that are infected with malicious software that lets criminals use them to send spam and launch denial-of-service attacks as part of extortion rackets.
Botnets or zombie networks are groups of computers that have been infected by malware that allow the malware to control the infected PC and use it to send spam or launch distributed denial of service (DDoS) attacks.
http://boingboing.net/2007_05_01_archive.html
http://weblog.infoworld.com/techwatch/archives/cat_security.html
0.41331

Wed Dec 31 16:00:00 PST 1969
Tue Jul 03 15:00:37 PDT 2007
New Scam Targets Bank Customers (Click for story) SANS Internet Storm Center is reporting on a new strain of IE Malware.
The prolific Storm malware is on the attack again, according to the folks at the SANS Internet Storm Center (ISC).
http://www.personal.psu.edu/faculty/w/r/wrp103/oldnews.html
http://www.us.first.org/newsroom/globalsecurity/
0.39362

Wed Dec 31 16:00:00 PST 1969
Wed Dec 31 16:00:00 PST 1969
Using malware or software designed to infiltrate a computer system, hackers steal account information for users of MMO games and then sell off virtual gold, weapons and other items for real money.

The computer then becomes part of a bot network, which can then be used to launch denial of service attacks, install keylogging software and steal personal account information and other malicious activities.
http://www.futurebrief.com/security2006.asp
http://weblog.infoworld.com/techwatch/archives/cat_security.html
0.38138

Wed Dec 31 16:00:00 PST 1969
Fri Oct 28 15:11:32 PDT 2005
Moreover, only 4 percent of the successful DoD attacks were noticed by network administrators, and only a small percentage of those detected were reported to authorities.
Thirty percent of respondents have no clue as to how many attacks their network was subjected to in the past year, and 22 percent do not know how many successful attacks transpired at that time.
http://www.ecommerce-guide.com/news/trends/print.php/7761_504441
http://www.umsl.edu/%7esauter/spam/index2.html
0.36669

Automatically Tracing Information Flow of Vulnerability and Cyber-Attack Information through Text Strings

Neil C. Rowe, Eric J. Sjoberg,

and Paige H. Adams

U.S. Naval Postgraduate School

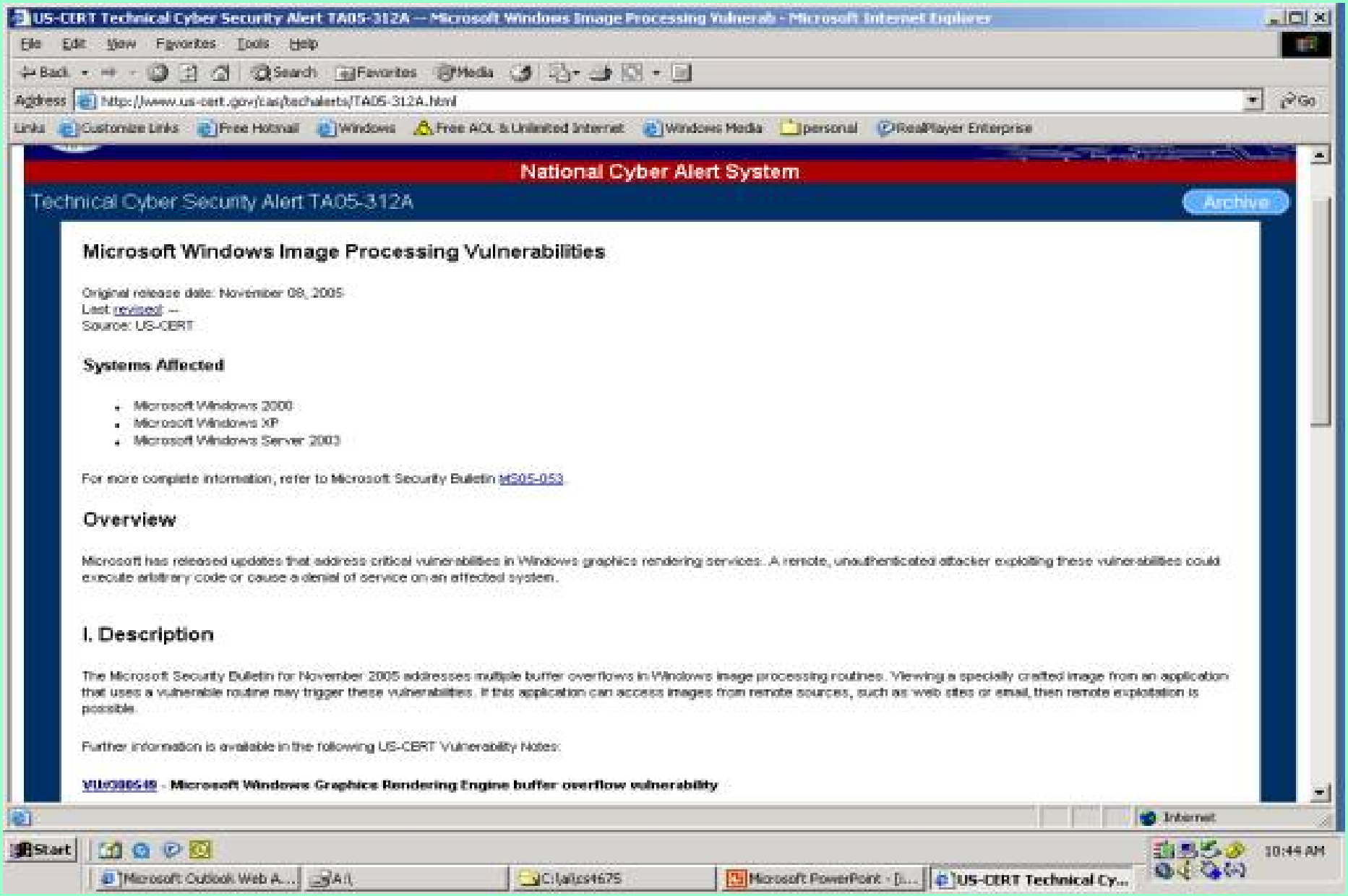
ncrowe@nps.edu

June 2008

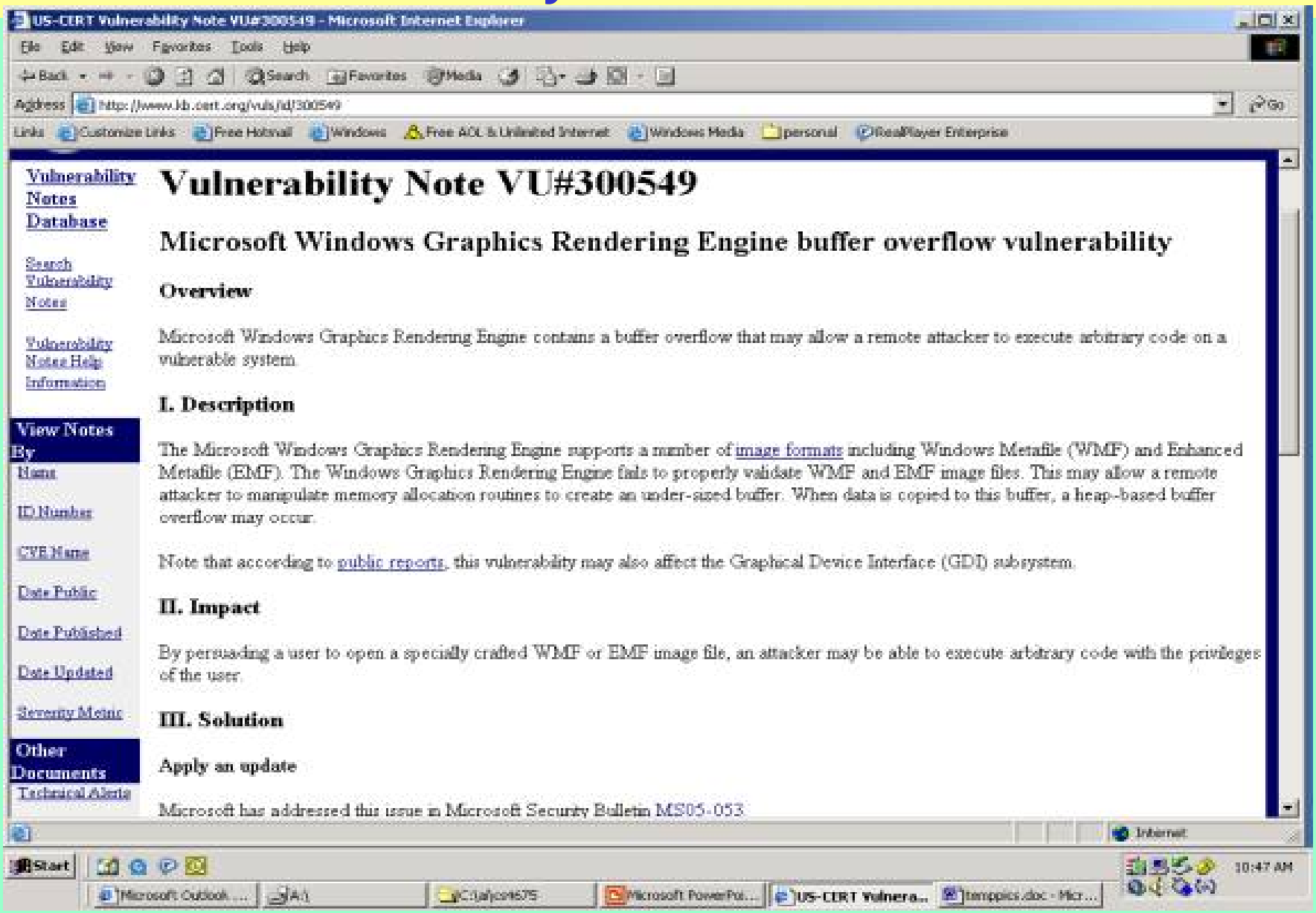
Overview

- Quick defenses to new cyberattacks are critical.
- Vulnerability and attack warnings get disseminated through several CERT sites, MITRE sites, vendor sites, etc.
- But there's little systematic analysis of this information dissemination:
  - Who's copying who, and who originates defense?
  - How fast can a new attack be handled today?
  - Are there bottlenecks in information dissemination?
- We are developing data-mining techniques to analyze this data.
- CVE numbers help track the same vulnerability, but are only used on the most formal pages.
- We are also starting to correlate vulnerability information with observed alerts and packets.

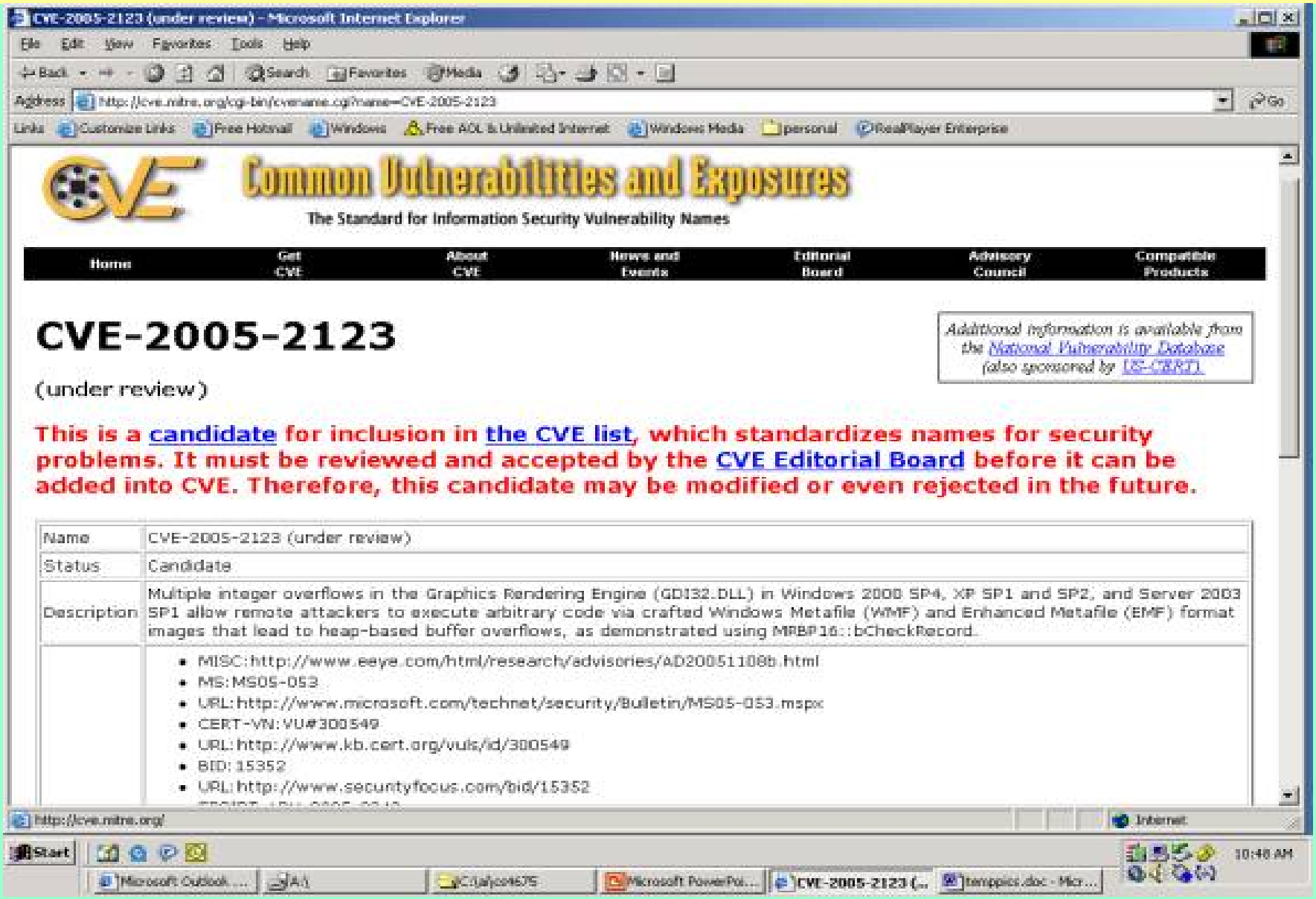
CMU-CERT provides general info about vulnerabilities



CERT Vulnerability Notes have more details

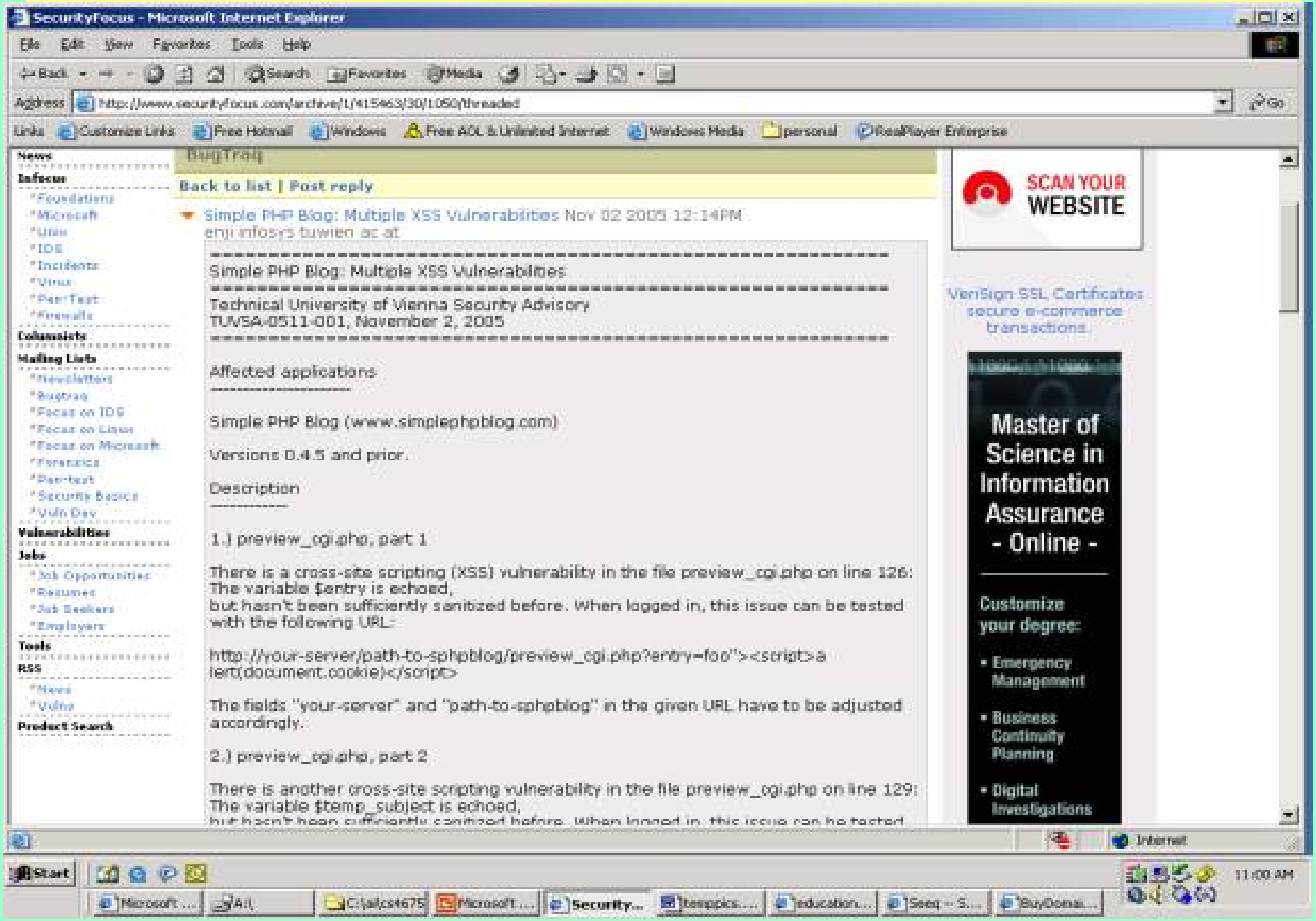


CVE provides index numbers on vulnerabilities

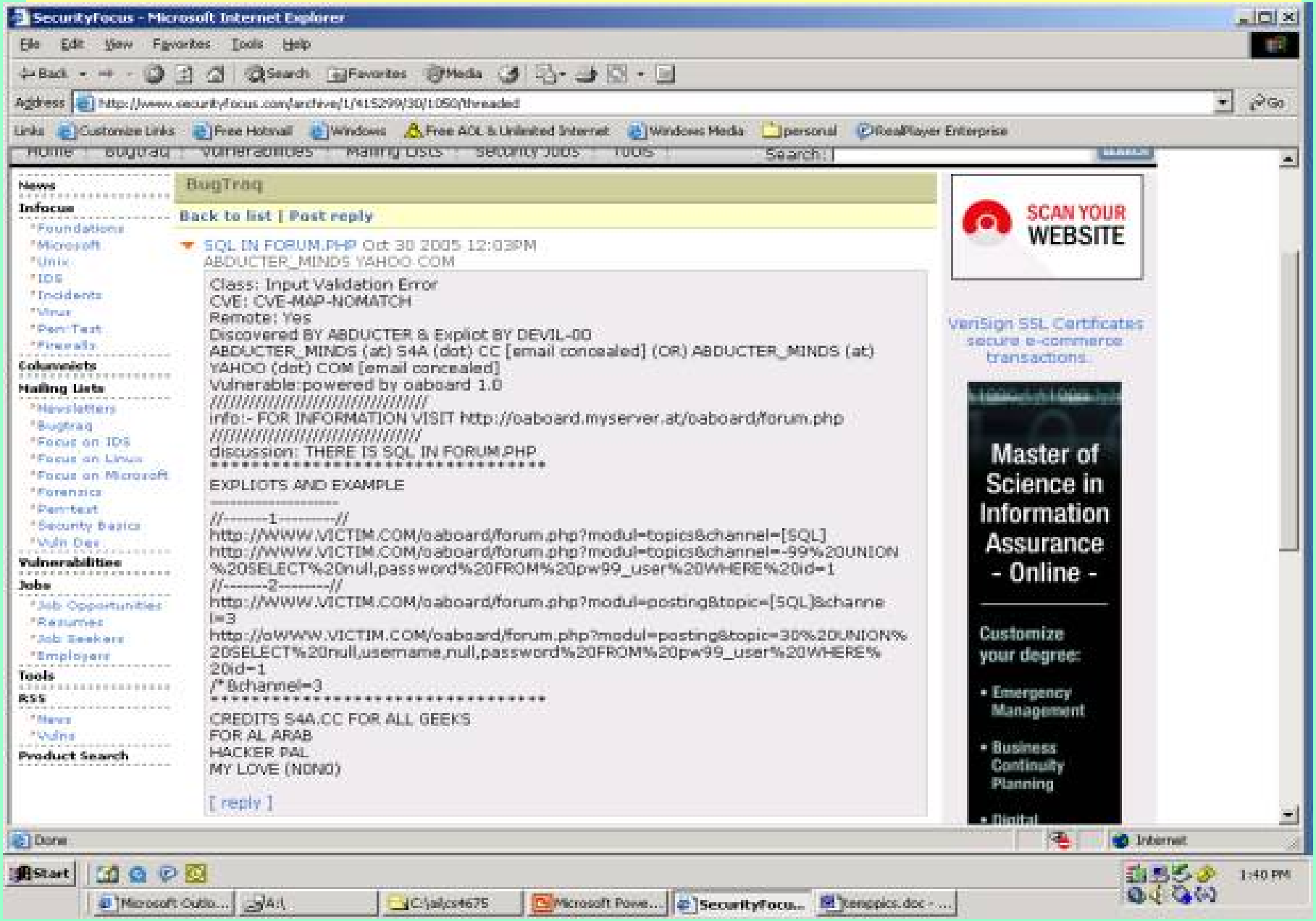


Security Focus / Bugtraq collects news on vulnerabilities

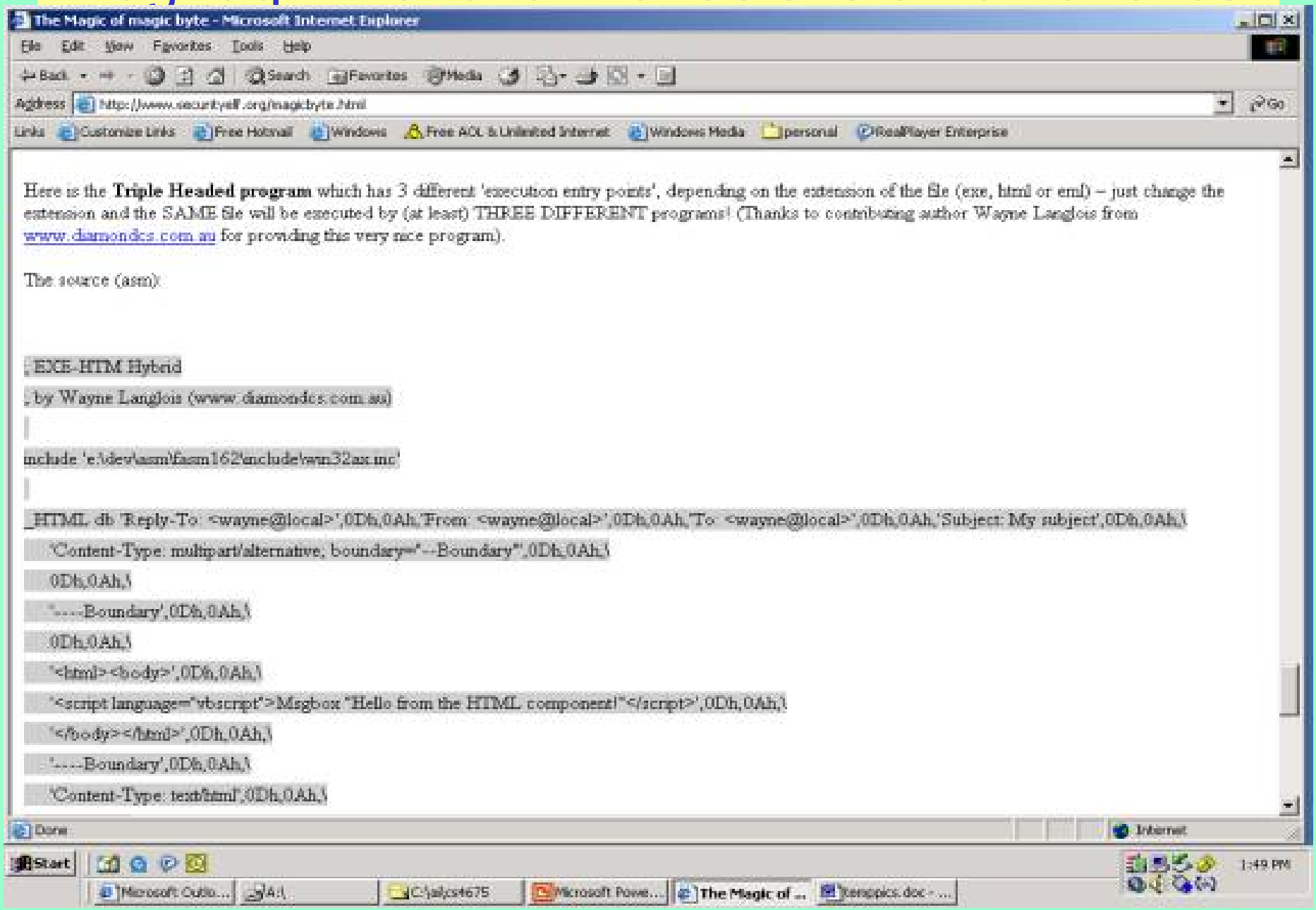
Example Bugtraq vulnerability description



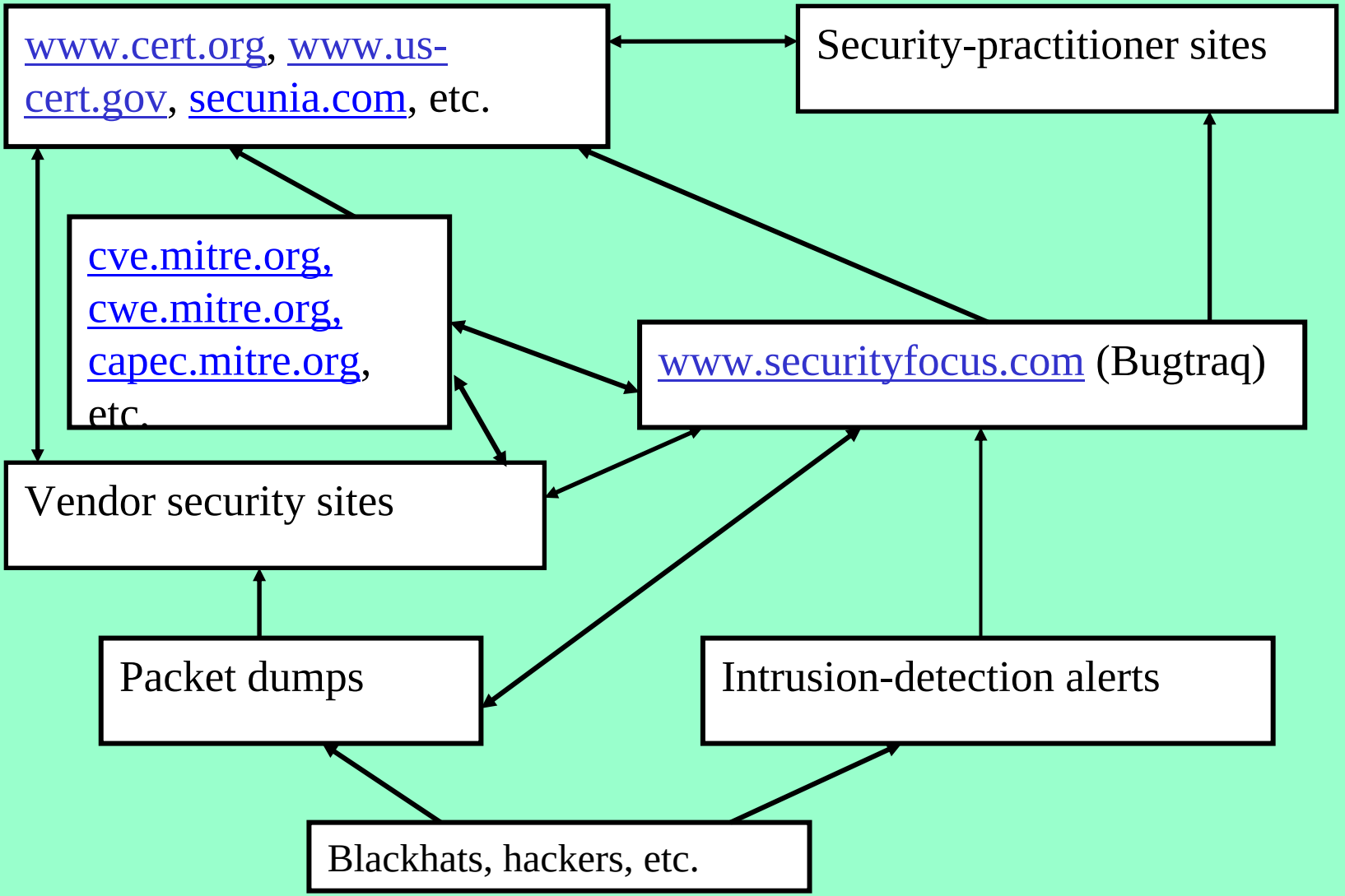
Bugtraq description example, second screenful



Bugtraq links to further details on other sites



Major data flows of vulnerability information



First approach to tracking flow of information security information: Keyword matching of Web pages

- Collect security keywords related to alerts.
- Send a subset of them to a browser site like Google.
- Collect the URLs of the top matches.
- Retrieve the pages of the top matches.
- Compare words of each sentence on a page to each sentence on another page.
- Exclude structural “stop words” (e.g. “the”, “in”, “then”, “system”).
- Find all very-close matches between sentences.

Classification of exact sentence matches

- Normal routine copying of pages, as when a Web site collects important papers on security.
- Common authorship on sites, e.g. www.demboo.info/Carbon-cheats.htm and www.mulax.info/Sims-cheats.htm both say “For walkthroughs, cheats and tips call 09067 53 54 55."
- Acknowledged citation, particularly if the text is quoted or indented. This can be distinguished by words such as "says“ and "stated".
- Plagiarism.
- "Boilerplate", formalized statements for some legal or policy objective, e.g. "Additional Information: For the most up-to-date information regarding these vulnerabilities, please visit the CERT/CC Vulnerability Notes Database at: http://www.kb.cert.org/vuls/".

Classification of inexact sentence matches

- Common authorship, e.g. "IGN is the ultimate Spider-Man: The Movie resource for trailers, screenshots, cheats , walkthroughs" versus "IGN PS2 is the ultimate resource for PlayStation 2 trailers, screenshots, cheats, walkthroughs".
- Acknowledged citation.
- Acknowledged paraphrase, e.g. "According to the US-CERT there is publicly available exploit code for multiple vulnerabilities in Sun Java Runtime Environment (JRE)" .
- Unacknowledged paraphrase.
- Boilerplate, e.g. “Further information is available in the following US-CERT Vulnerability Note".
- Accidental similarities, e.g. "An attacker could use a specially crafted web page to exploit the vulnerability and take control of a system, warned Danish security firm Secunia" and "An attacker could exploit these vulnerabilities by using specially crafted network traffic, by convincing you to click on a specially crafted URL, or by convincing you to open a specially crafted Office document".

Calculate similar sites from similar sentences

For keywords “vulnerability”, “ICMP”, “packets”, “flags”, and “footprinting”:

34.966: www.ecst.csuchico.edu to www.yolinux.com

34.943: www.e-infomax.com to www.ecst.csuchico.edu

34.943: www.ecst.csuchico.edu to www.uni-kiel.de

34.943: www.linuxdig.com to www.ecst.csuchico.edu

31.713: docs.mandragor.org to www.ecst.csuchico.edu

31.696: www.ecst.csuchico.edu to www.arameya.com

28.713: www.e-infomax.com to www.uni-kiel.de

28.713: www.linuxdig.com to www.e-infomax.com

28.713: www.linuxdig.com to www.uni-kiel.de

28.710: www.e-infomax.com to www.yolinux.com

28.710: www.linuxdig.com to www.yolinux.com

28.710: www.uni-kiel.de to www.yolinux.com

28.426: www.cs.wisc.edu to www.ecst.csuchico.edu

27.397: docs.mandragor.org to www.arameya.com

26.796: docs.mandragor.org to www.e-infomax.com

26.796: docs.mandragor.org to www.linuxdig.com

26.796: docs.mandragor.org to www.uni-kiel.de

26.793: docs.mandragor.org to www.yolinux.com

26.793: www.e-infomax.com to www.arameya.com

26.793: www.linuxdig.com to www.arameya.com

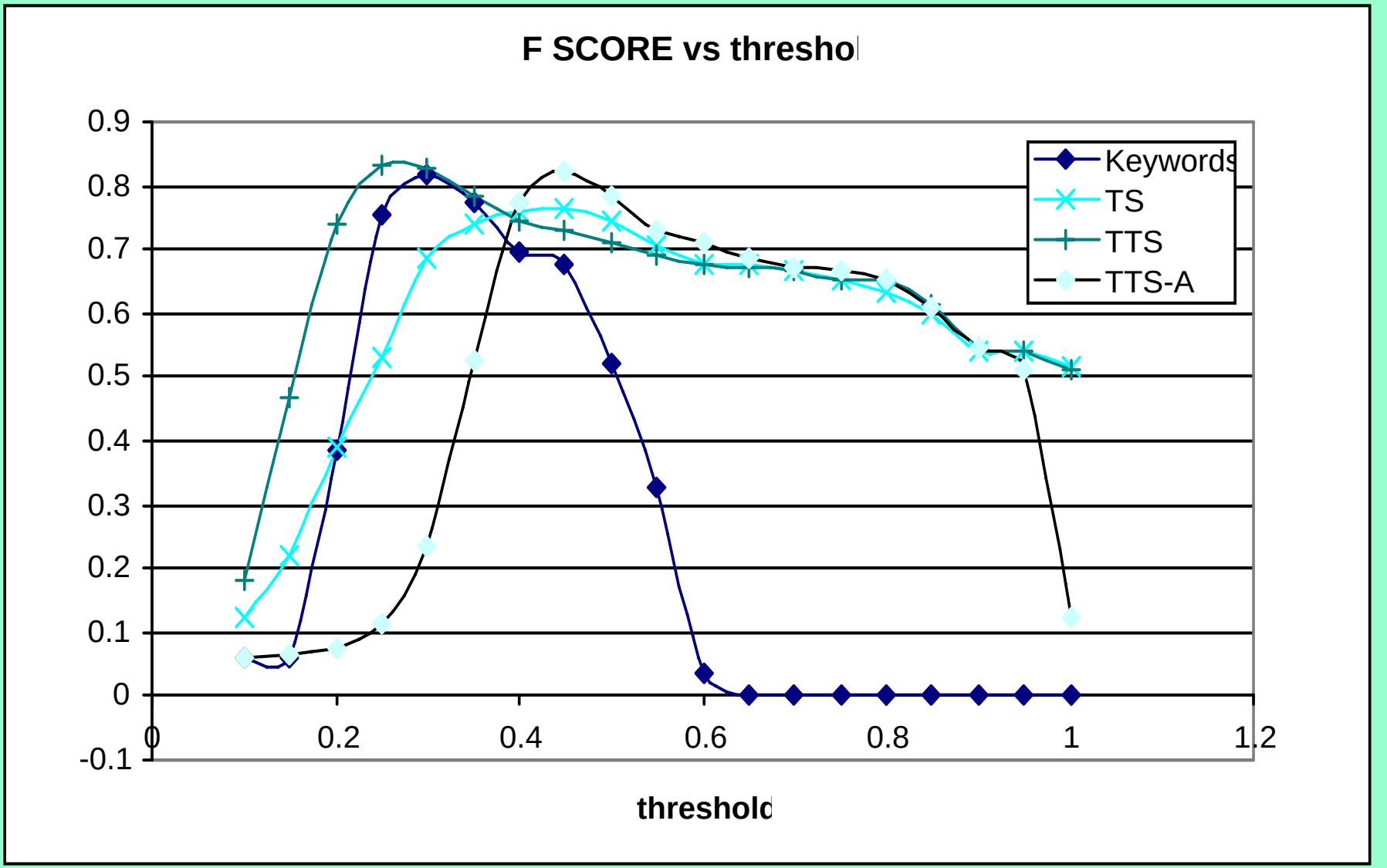
26.793: www.uni-kiel.de to www.arameya.com

26.788: www.arameya.com to www.yolinux.com

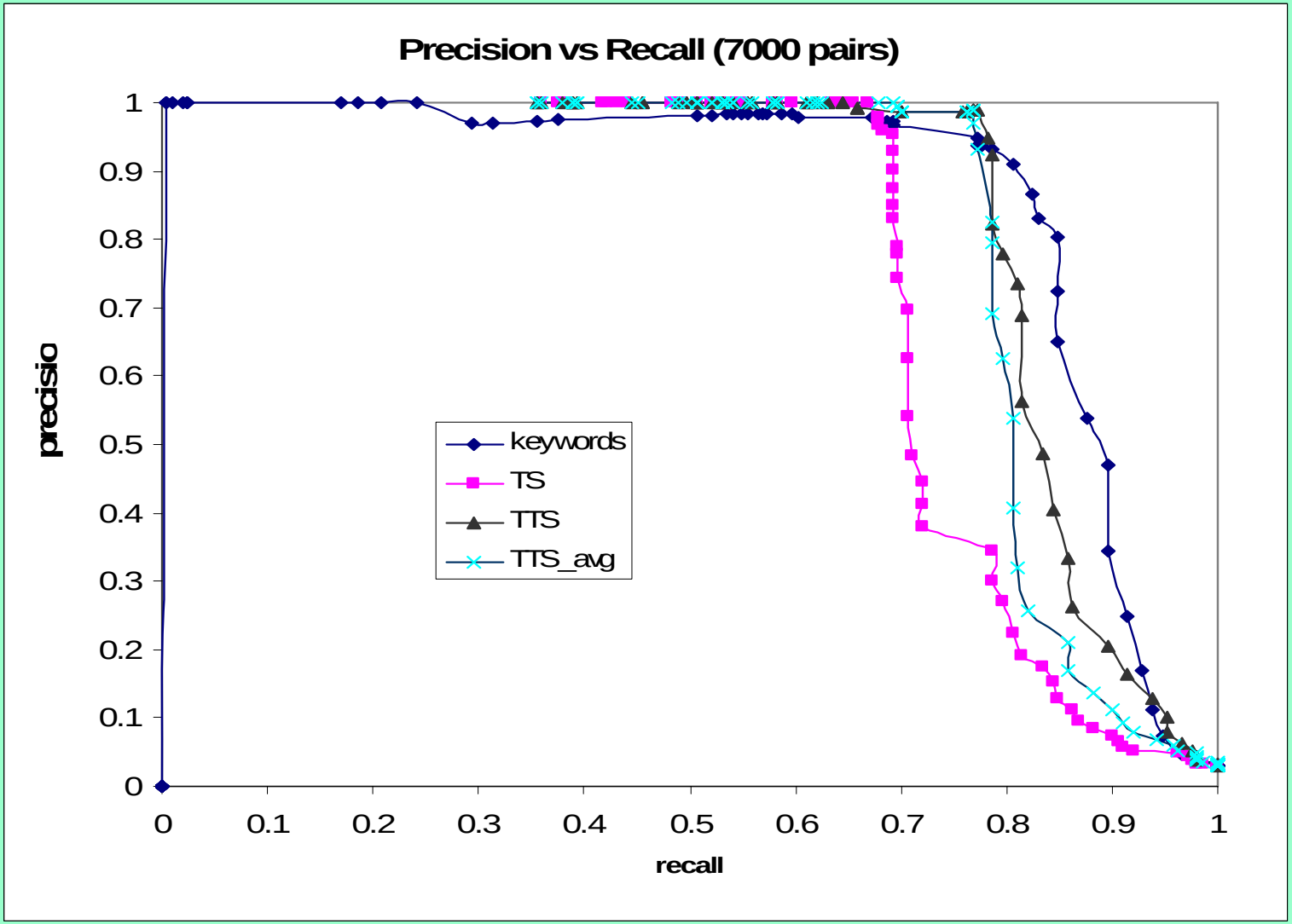
Deeper analysis of page similarity

- We can do better on comparing sentences if we know the parts of speech used.
- This suggests using a tagger (we used the Brill one).
- We compared (1) matching keywords only, (2) matching tags only (ignoring tag order), (3) matching both keywords and tags, (4) extending 3 to include bigram matches.
- Methods (3) and (4) performed the best, but (4) was not significantly better than (3), hence bigrams are not necessary for good performance.

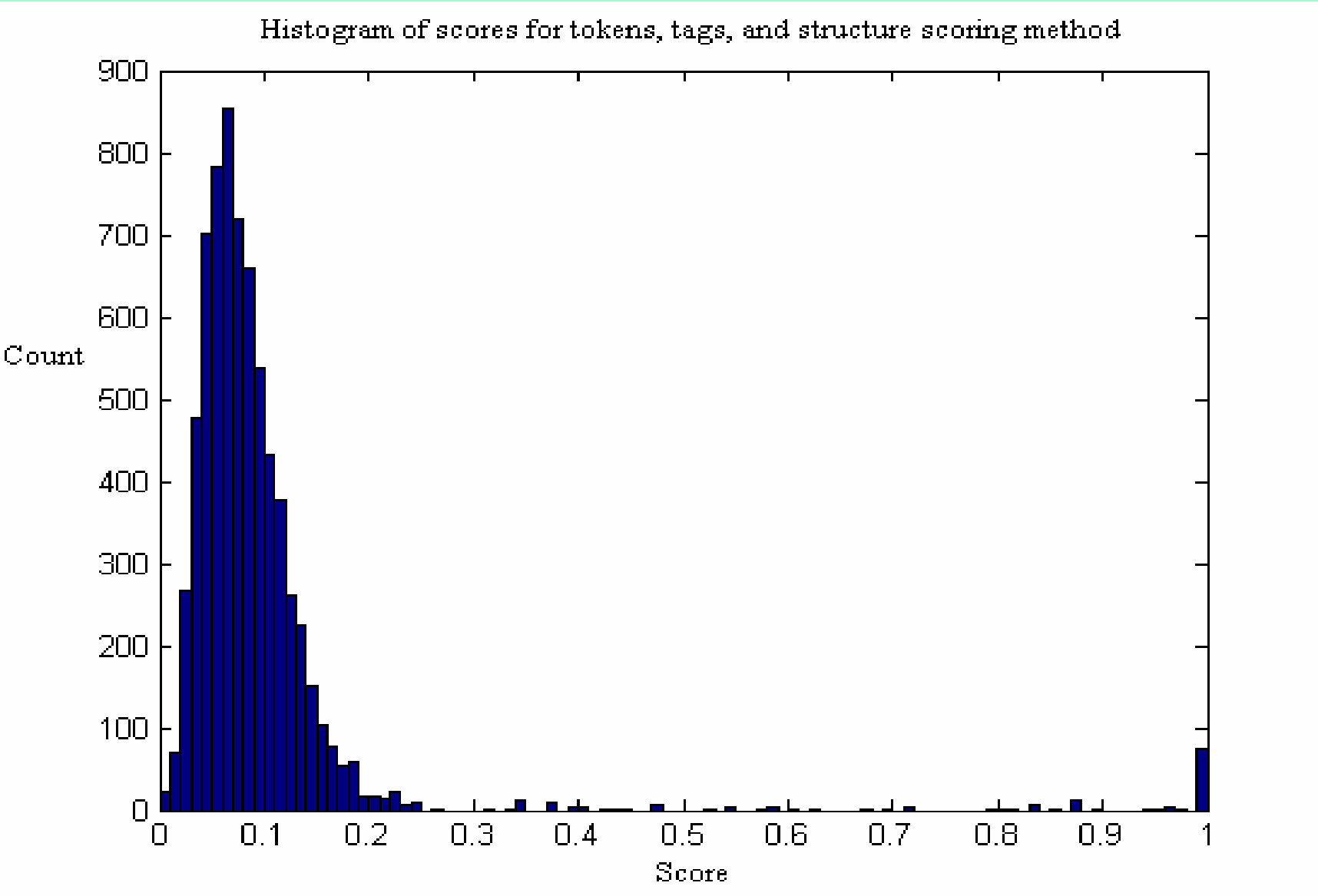
Performance is better with tagging



Another view of the data



The distribution of random sentence matches



Match failures suggest future directions

"One strain of scam email makes the bogus claim that recipients have won one of the much sought after devices in a bid to trick prospective marks into visiting a malware loaded site."

"Email recipients are sent a bogus email informing them that they have won a new iPhone, in reality the email contained malware designed to subvert and compromise the user's computer."

"NanoScan is a rapid, light scanner that currently detects over 750,000 active viruses, spyware, Trojans and other malware within just one minute."

"Panda Software has launched the mini, customisable version of NanoScan, the instant virus scanner from Panda Software, designed to detect active malware on a PC in less than one minute."

"According to a 2005 FBI Cyber Crime Study, 90 percent of small businesses had at least one cyber security incident within the past year."

"In fact, of the 500 companies that responded to a recent FBI survey, 90 percent said they'd had a computer security breach, and 80 percent of those said they'd suffered financial loss as a result."

"Using ideological attraction, the Soviets successfully recruited many high-level spies."

"At that time period the Soviets recruited their spies using ideological motivation."

Using more direct attack data

- Alerts and vulnerability notes are secondary-source information.
- More direct information would be the output of intrusion-detection systems.
- We can pull text related to specific alerts that are noticed on a system and find attack trends.
- Still more direct are packets themselves.
- Many of these contain text strings that can suggest attack trends.

Example intrusion-detection system text

Consider Snort alert from our honeypot: *Date: 2007-09-12 Time: 15:46:56.148-07 Alert_code: 1394 Alert_description: SHELLCODE x86 NOOP IP_address_1: 89.26.217.22 Port_#_1: 4310 IP_address_2: 192.168.0.3 Port_#_2: 445 time_to_live: 118*

This is co-referenced with the text:

- **Message** SHELLCODE x86 NOOP
- **Summary** This event is generated when an attempt is made to possibly overflow a buffer. The NOOP warning occurs when a series of NOOP (no operation) are found in a stream. Most buffer overflow exploits typically use NOOPs sleds to pad the code.
  Impact This might indicate someone is trying to use a buffer overflow exploit. Full compromise of system is possible if the exploit is successful.
  Detailed Information This rule detects a large number of consecutive NOOP instructions used in padding code. It's not specific to a particular service exploit, but rather used to try and detect buffer overflows in general. It is common for buffer overflow code to contain a large sequence of NOOP instructions as it increases the odds of successful execution of the useful shellcode.
  Affected Systems Any x86 programs.
  Attack Scenarios An attacker uses a buffer overflow exploit which contains the following payload: 90 90 90 90 90 90 90 90 90 90 /bin/sh
  Ease of Attack Simple.
  False Positives High, This event may be generated by applications such as ftp and http when binary data is being transferred. A false Positive can be generated if the snort sensor detects text from an IRC client or any other application that passes data plaintext. The event is generated if Snort detects several (a) characters in a row - such as 'aaaaaaaaaa'.

Example packet text strings

```
09/14-00:47:21.626361 131.120.18.41:53 -> 192.168.0.3:3559 UDP
   TTL:111 TOS:0x0 ID:15349 IpLen:20 DgmLen:145 Len: 117
47 59 81 83 00 01 00 00 00 01 00 00 02 67 63 06  GY.............gc.
5F 6D 73 64 63 73 08 55 53 4E 42 41 52 4F 4E 05  _msdcs.USNBARON.
6C 6F 63 61 6C 00 00 06 00 01 00 00 06 00 01 00  local...........
00 00 00 00 40 01 41 0C 52 4F 4F 54 2D 53 45 52  ....@.A.ROOT-SER
56 45 52 53 03 4E 45 54 00 05 4E 53 54 4C 44 0C  VERS.NET..NSTLD.
56 45 52 49 53 49 47 4E 2D 47 52 53 03 43 4F 4D  VERISIGN-GRS.COM
00 77 A1 C8 65 00 00 07 08 00 00 03 84 00 09 3A  .w..e..........:
80 00 01 51 80                                   ...Q.

09/16-22:43:13.038582 131.120.18.41:53 -> 192.168.0.4:1052
UDP TTL:111 TOS:0x0 ID:13512 IpLen:20 DgmLen:137 Len: 109
FF FA 81 83 00 01 00 00 00 01 00 00 08 64 6F 77  .............dow
6E 6C 6F 57 64 0D 77 69 6E 64 6F 77 73 75 70 64  nloWd.windowsupd
61 74 65 03 63 6F 6D 00 00 01 00 01 C0 15 00 06  ate.com.........
00 01 00 00 0E 10 00 35 03 6E 73 31 04 6D 73 66  .......5.ns1.msf
74 03 6E 65 74 00 06 6D 73 6E 68 73 74 09 6D 69  t.net..msnhst.mi
63 72 6F 73 6F 66 74 C0 23 77 A1 A5 3D 00 00 03  crosoft.#w..=...
84 00 00 02 58 00 09 27 C0 00 00 03 84           ....X..'.....
```

The “MARB MEOW” strings

```
05 00 00 03 10 00 00 00 A8 06 00 00 E5 00 00 00  ................
90 06 00 00 01 00 04 00 05 00 06 00 01 00 00 00  ................
00 00 00 00 32 24 58 FD CC 45 64 49 B0 70 DD AE  ....2$X..EdI.p..
74 2C 96 D2 60 5E 0D 00 01 00 00 00 00 00 00 00  t,..`^..........
70 5E 0D 00 02 00 00 00 7C 5E 0D 00 00 00 00 00  p^......|^......
10 00 00 00 80 96 F1 F1 2A 4D CE 11 A6 6A 00 20  ........*M...j. 
AF 6E 72 F4 0C 00 00 00 4D 41 52 42 01 00 00 00  .nr.....MARB....
00 00 00 00 0D F0 AD BA 00 00 00 00 A8 F4 0B 00  ................
20 06 00 00 20 06 00 00 4D 45 4F 57 04 00 00 00   ... ...MEOW....
A2 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  ...............F
38 03 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  8..............F
00 00 00 00 F0 05 00 00 E8 05 00 00 00 00 00 00  ................
01 10 08 00 CC CC CC CC C8 00 00 00 4D 45 4F 57  ............MEOW
E8 05 00 00 D8 00 00 00 00 00 00 00 02 00 00 00  ................
```

Conclusions

- Information transmission is important with security alerts and other security intelligence.
- The transmission infrastructure has been built informally – and it mostly works.
- But there can be bottlenecks and redundancies.
- And an adversary could exploit weaknesses or attack the infrastructure itself.
- We need data on the infrastructure to make good decisions about it.
- We’re just starting this research.