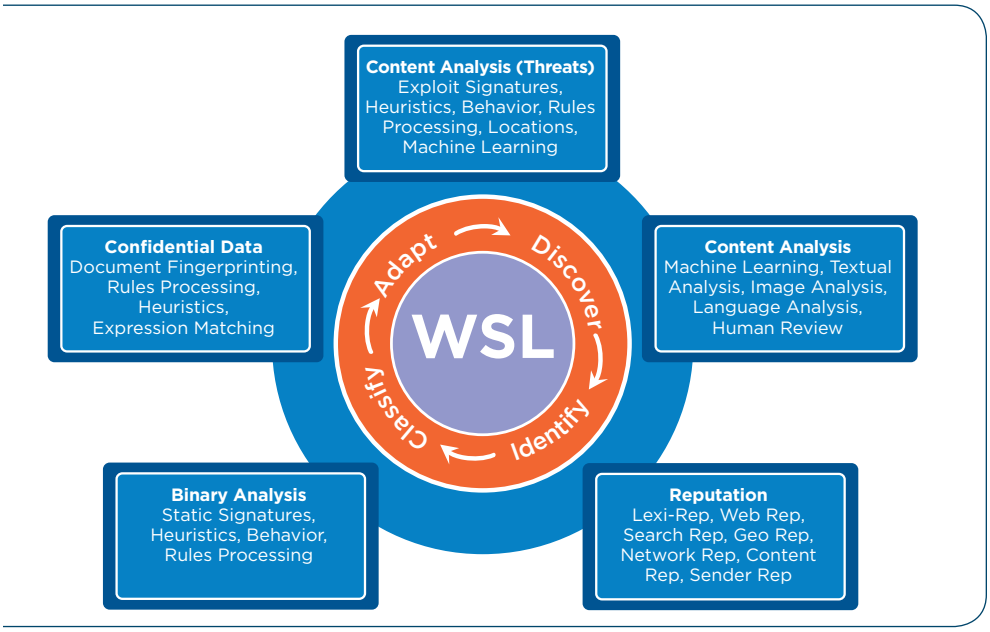
A Websense® Technical White Paper

Websense Content Research Cortex

websense®
ESSENTIAL INFORMATION PROTECTION™

Classifying content is what Websense® is all about. More than 100 researchers worldwide apply a vast array of classification techniques to block malicious or unwanted data from entering the network and protect confidential or proprietary data from leaving. Behind the scenes, they have built the necessary tools, data and infrastructure into a web of interconnected technologies known as the Content Research Cortex. In tandem with the vast data collection capabilities of the ThreatSeeker™ Network, these technologies allow Websense to discover, identify, classify, and adapt to content trends on a global scale. This paper provides an insider's perspective into how Websense has invested in the necessary research, tools, and people to continuingly evolve the Content Research Cortex to meet the challenge of content classification and protection.



Websense Content Research Cortex

1 Introduction

Over fifteen years ago, simple filters were used to categorize and block access to Web sites with objectionable content. A small dedicated research team encoded a list of the sites and attached a category. A growing customer base helped find the ones they missed and Web filtering was born.

Today, the World Wide Web is several million times bigger and evolving at a dizzying pace. Applications that used to be separate—email, instant messaging, and desktop applications—are now packaged as Web traffic, too (in the form of Webmail and Web 2.0 gadgets). Static content from a single source has been supplanted with a torrent of script, advertisements, and user-generated content from a myriad of generally anonymous sources. Confidential information inside the corporate network has found its way into the public domain or an attacker's hands through a rapidly increasing number of channels. Attacks from both inside and outside the network are far more sophisticated and frequent. Suffice to say, it's a lot harder than it's ever been for organizations to protect their proprietary information and ensure Essential Information Protection™.

Classifying content is what Websense is all about. More than 100 researchers worldwide apply a vast array of classification techniques to block malicious or unwanted data from entering the network and prevent confidential or proprietary data from leaving. Behind the scenes, they have built the necessary tools, data, and infrastructure into a Web of interconnected technologies known as the Content Research Cortex. In tandem with the vast data collection capabilities of the Websense ThreatSeeker™ Network, these technologies allow Websense to discover, identify, classify, and adapt to content trends on a global scale. This paper provides an insider's perspective into how Websense has invested in the necessary research, tools, and people to continuingly evolve the Content Research Cortex to meet the challenge of content classification and protection.

2 Discovery

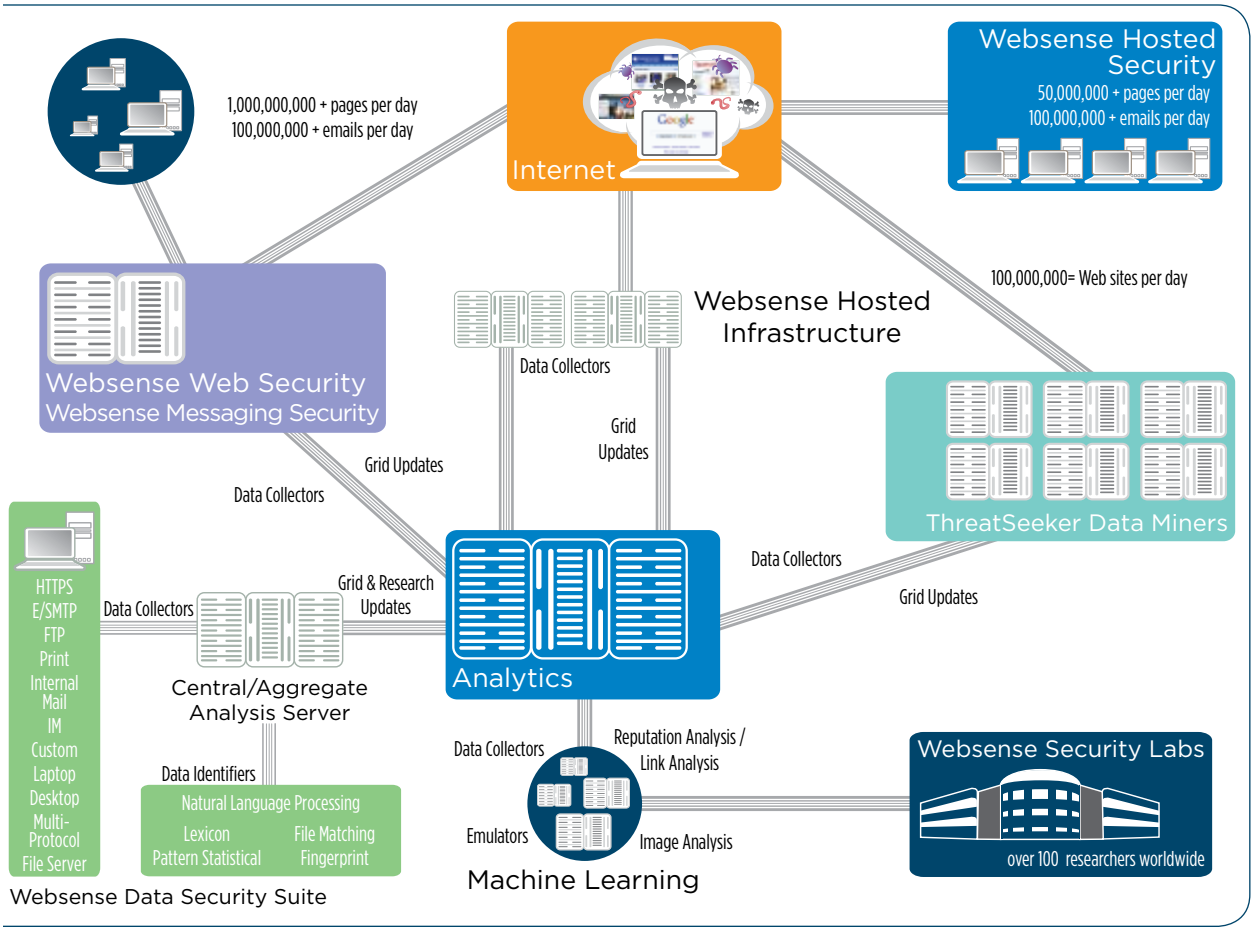
The process of finding the up-to-the-millisecond content in the real-world is called *discovery*. For an organization, it encapsulates both the massive store of mostly unknown, untrusted, inbound content that must be filtered, and the internal network with its stores of proprietary and confidential information that must be guarded. Some forms of inbound data discovery are specific, such as finding a new or recently compromised page that is hosting malicious code or a confidential document in an unexpected place. Other forms are more general, including tracking spikes in data traffic patterns, defining a host or sender reputation across a number of characteristics, or classifying a content ecosystem. On outbound data, discovery includes finding all the system's endpoints (e.g. desktops, laptops, servers) in an organization, examining the data to determine what kinds of information are confidential or at risk of being misused or leaked, and providing a clear and concise map of the sensitive organizational information to ensure its protection.

Given the size and volatility of the Internet, effective discovery today requires a mountain of data collection and an army of data collectors. Malicious or objectionable content could be anywhere—in well-known, high-traffic areas or in brand new domains that appear and disappear within a few days.

Another requirement for effective discovery is to be able to hone the data collection and harvest only the general trends or interesting phenomena. There's far too much data out there—well over an exabyte (1,000,000,000,000,000,000 bytes)—to monitor, transmit to Websense, and store for any sort of real-time analysis. Instead, data collectors must have a concise set of interesting things that they monitor and harvest. Similarly, they must support a rich language to switch gears and dynamically monitor a wide range of new content trends. And discovering data once is rarely enough. For Web pages, executable files and a host of other content types, the data must be re-checked repeatedly to make sure no unwanted changes have taken place.

The Websense Internet HoneyGrid™ is the main Websense technology to meet the challenge of discovery on this massive scale. Over 50 million systems are interconnected in an adaptive feedback network that scans all incoming and outbound data, probes for interesting or new content, and sends feedback to Websense (or to the local administrator for data leakage protection). Updates to probes and/or protections are sent out to all systems in the ThreatSeeker Network every few minutes.

For a more detailed account of this adaptive discovery network and its underlying technologies, read "The Websense ThreatSeeker Network: Leveraging Websense HoneyGrid Computing [WEB08-A]."



2.1 Discovery Challenge: Drive-By Rush Hour

Imagine hundreds of thousands of legitimate Web sites compromised nearly simultaneously with a new JavaScript injection attack. The script loads content rife with vulnerabilities in an attempt to install malware that will compromise the local host and steal confidential data. How can you determine what's happening in real-time while simultaneously protecting the hundreds of thousands of vulnerable domains?

First, the Websense Internet HoneyGrid can probe for this form of attack. Most mass attacks are delayed variations on a single theme whose root cause points back to a known (and generally recently discovered) vulnerability. The probe encodes detection for this root cause to catch all variants of the attack and is distributed to the millions of systems on the HoneyGrid. When the mass attack hits the wire, those systems are protected as they begin to visit the compromised sites and simultaneously send intelligence about the compromised page back through the ThreatSeeker Network. After receiving overwhelming intelligence that a mass attack is under way, the ThreatSeeker data miners begin mining the compromised sites to find additional sites that the HoneyGrid clients have not yet visited. By virtue of this massive scale distributed discovery network, the vast majority of sites are flagged within minutes of compromise.

This mass compromise scenario and Websense's discovery is exactly what happened on April 22, 2008 [WEB08-C].

2.2 Discovery Challenge: The Pink Elephant You Can't See

Imagine a marketing manager has a customer list, including contact and sales detail, saved on a network drive and locally on her laptop for use when she travels. The employee is working hard to complete a project before she goes on vacation, but the project runs over and she is forced to take some work with her. Rather than take her bulky laptop, she emails the customer list to her Gmail account and saves the downloaded list to the computer in the hotel's business center. How do you know any of this happened?

Websense Internet HoneyGrid can discover the confidential customer list stored on the network and laptop of the marketing manager. This discovery is an important step in mitigating risk. Confidential information stored in potentially unsafe locations, like on a laptop, is at risk of being lost. In this example, Websense would have provided visibility into what data was stored in risky locations, and provided a means of redress to secure the data. Additionally, Websense, using Websense Data Security Suite, could have authorized the marketing manager to use the customer list, but prohibited her from sending it outside the organization (i.e., to Gmail). If the marketing manager had brought her laptop, Websense could have known that the file was stored locally and forced file-level encryption (with third-party integration) to ensure the file was secured. In addition, Websense could have prohibited pre-specified policy controls such as blocking of local printing, copy and paste, or copy to a USB device.

In the end, it's easy to imagine the controls that could have been in place to secure the customer list; however, what was required was the ability to discover where the data was and how it was being used.

3 Identification

Once a new or interesting piece of data has been discovered, the process of identification begins. The identification process transforms data from its original form (essentially a collection of zeros and ones) into a collection of characteristics that can be understood by all Websense technologies and later classified. This is no easy task. To begin with, all data is packaged in a particular format or protocol, depending on the purpose of the content. Readable text could be presented in one or multiple natural human languages from a selection of hundreds. Some content is further masked by encryption or intentional obfuscation. Each of these containers must be stripped away to reveal the true nature of the content and begin the process of classification.

3.1 Protocols

A protocol is "a convention or standard that controls or enables the connection, communication, and data transfer between two computing endpoints [WIK08-A]." For example, unencrypted Web traffic arrives in the Hypertext Transfer Protocol (HTTP), encrypted Web traffic in the Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) and email is generally sent using the Simple Mail Transfer Protocol (SMTP). Every application today (e.g. instant messaging, file transfer, database) uses some sort of protocol for its data transmission.

While protocol wrapper information can be inherently interesting for classification, in some cases, proper classification requires that the protocol format be understood to get at the nature of the data underneath. To that end, Websense has created the Websense Network Agent, a technology that can unwrap and inspect the content from well over 100 of the most common protocols. By stripping away the transport layers and understanding the content stream underneath, Websense can apply a universal set of analysis techniques to the data regardless of how it was transmitted, including traffic over Web, email, instant messaging, file sharing networks, peer-to-peer networks and many others. For a complete list, see Appendix A.

3.2 Content Types

Once the network transport layers have been stripped, most information today is still not communicated as a blob of human-readable text. Different kinds of content require different actions or presentations, and that meta-information is stored using a variety of content types. A content type is basically a pre-defined data format that specifies which applications understand the underlying content and provide a standard way for that content to be accessed. In order to properly classify content for these formats, Websense must be able to efficiently and accurately navigate them. In numerous cases, content type formats can be intentionally violated by attackers to target critical security vulnerabilities and exploit remote systems [MIC08-A, MIC08-B]—these must also be carefully scrutinized.

Websense supports hundreds of content groups and types. A few examples of well-known content groups and types include:

- Documents (Microsoft Office/OLE, Adobe/PDF)
- Multimedia (JPEG, GIF, BMP, AVI, WAV, Flash)
- Executables (Win32 EXE/DLL/ActiveX, Linux ELF)
- Web (HTML, XML, JavaScript)

3.3 Encryption, Compression & Obfuscation

Sometimes a legitimate application (or an attacker's creation) will go to extra effort to hide the nature of content. These techniques fall mainly into three categories: encryption, compression, and obfuscation.

3.3.1. Encryption

Encryption is "the process of transforming information (referred to as plain text) using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a "key" [WIK08-B]. One legitimate example of encryption is that Web applications traditionally use a user-entered login and password for a key; once authenticated, encrypted data can then be transmitted using the HTTPS protocol. This remains a potential attack vector since hacked Web sites can deliver malicious content over any protocol. Websense thwarts these encrypted attacks by providing the capability for SSL decryption at the gateway.

Illegitimate programs are encrypted by attackers to hide their content. Known malware of this type is identified with a variety of fingerprinting techniques. Unknown binaries are emulated, decrypted, and inspected by Websense analytics. The malicious ones are classified and updated to all products automatically every few minutes.

3.3.2. Compression

A number of well-known compression tools legitimately use standard encoding formats (e.g. ZIP, RAR) to shrink data before transmission. Attackers can also use these same tools to re-encode and therefore hide the actual contents underneath. As a consequence, Websense supports over one hundred compression formats to inspect the data underneath. Packers are a special form of compression tool generally reserved for applications to reduce their size. Outside attackers or insiders trying to hide their activities have produced hundreds of customized packers, each subtly different from the last, to mask their malicious code from security tools. Nicolas Brulez, a leading researcher in the area, devised and implemented an innovative approach for both generic and specific application unpacking that can discover and defeat this tactic. For more information and some interesting extensions to it, read Joren McReynold's report on "Packer Detection and Generic Unpacking Techniques [MCR08]."

3.3.3. Obfuscation

Obfuscation (also called polymorphism or metamorphism) is another mechanism used by attackers to bypass traditional fingerprinting techniques on malicious code. It takes advantage of the fact that there are literally an infinite number of ways to represent even the simplest constructs in code. A very simple example is that the number "0" can be represented arithmetically as "1 minus 1", "2 times 0," '4 plus 1 minus 5," and so on. If the attacker replaces all of the numbers, strings, and other simple operations with one of its analogs each time a user downloads script or code, no fingerprinting technique will ever catch all of the variations. Websense has developed automatic

deobfuscation for all of the most common techniques in use today. For more information on these approaches, read "Unscrambling Custom Obfuscation and Executable Infection [BRU08]" and "Automated JavaScript Deobfuscation [CHE07]."

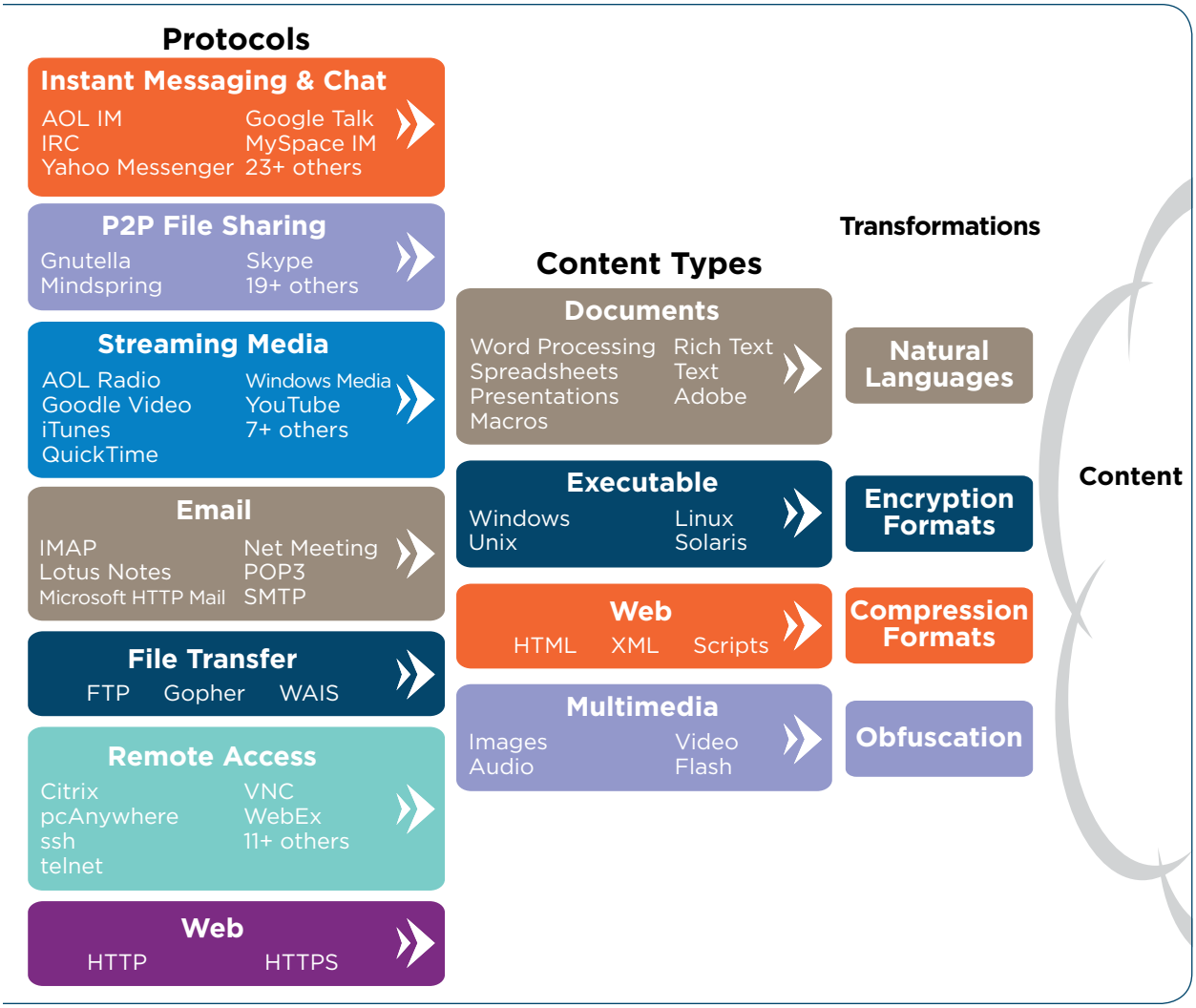
For email and data leak prevention, text obfuscation is another common practice to evade content filters. There can be a nearly limitless number of spellings and character replacements for a word, and each new piece of content could be generated with a different variation. For example, the word "great" might be switched with human-identifiable versions such as "gr8t", "grate" or "gr eat." Content classifiers must reconstruct the core content from these variations. With both program and text, the mere presence of obfuscation is generally a tip-off that something is amiss and can be used as a hint for deeper analysis. These techniques are most useful for spam detection in Websense messaging products and identifying malicious concealment of confidential data by Websense Data Security Suite.

3.4 Language

Natural language is often overlooked in content classification. Part of the reason for this is that traditional security and content filtering solutions speak the universal language of fingerprints and code. A URL is precisely what the name implies: a "Universal Resource Locator." A fingerprint is always a string of zeros and ones. Code is always a sequence of language-independent instructions.

This model breaks down when it is a requirement to identify the semantic nature of content. Language must be understood in order to determine if it is in violation to a zero-tolerance policy on objectionable content (e.g. pornography, hate groups) or even merely an unacceptable use of corporate resources (e.g. gambling, proxy avoidance, shopping). To divide global content into over 90 different semantic categories, you need natural language expertise.

Over 1 billion pieces of content from nearly every country around the globe are delivered to Websense Security Labs each day. Websense has language experts around the world that decipher more than 50 languages to determine the semantic nature of content (see Appendix B). With the help of these experts, researchers build fingerprints and train real-time classifiers to automatically classify unknown content at the gateway, in the cloud, and in the Websense Security Labs. (see Section 4.3.1 for more information on how it's done).



3.5 Identification Challenge: E Pluribus Unum

Imagine a worm that sends a link through email and a number of instant messengers simultaneously, including Skype, MSN Messenger, and ICQ (and contained skeleton code for Aim, Trillian, and Yahoo! Messenger, too). The target of the link is a packed, encrypted, program that compromises the local host and morphs itself with each new infection. The Trojan also steals information and begins sending out emails and links of its own to further propagate. How would you catch and prevent all of the variants across all of these attack vectors from infecting local hosts?

Websense protocol analysis would navigate the protocol information from each attack vector—Skype, ICQ, etc.—to identify the link and its target binary. Generic unpacking would decompress the file, and application virtualization network of systems (discussed in Section 4.6.3) would decrypt, analyze, and easily classify the binary as malicious. The essence of this single piece of content—a malicious binary—classifies all of the embedded links, emails, morphed binaries, and any other variations of this attack. Each would be summarily blocked and any required updates would be distributed to the entire Websense suite of products as needed within minutes.

Websense detected a multi-protocol worm attack, a variant of the Stration family, launched on May 23, 2007. [BOY07].

4 Classification

After the encryption, obfuscation, protocols, natural language permutations, and other data transformations have been stripped away, we can finally begin to classify the nature of the content underneath. This task is far from trivial. Content can be static (rarely changing) to dynamic (changing several times per second). Content can be created by one author (an email) or millions (Web 2.0 or social networking sites). Content can be socially engineered to be intentionally misleading (spam) or dangerous (remote execution exploits and Trojans). The most valuable content that enables business is frequently proprietary and confidential, but must be classified and secured from external access. The most dangerous kinds of content are creations by wily adversaries that attempt to bypass existing security controls. Trying to distinguish the intent of one kind of content from another can be a tricky business.

As a result, classifiers need to be tuned to both the flow of different technology streams as well as to the attackers that attempt to exploit them. Categorization techniques range from traditional static ones (such as fingerprints for known content) to the most general and dynamic (machine learning techniques for complex behaviors and trends that classify unknown content). It is this latter category that separates a brittle approach that is always chasing yesterday's news from a more robust one that reaches into the future to prevent unknown attacks. These classification techniques that work together to form the backbone of the core technology that is shared by all Websense products.

4.1 Fingerprints

A *fingerprint* is a sequence of bytes that can be found in a chunk of data. A large fingerprint might encompass the entire piece of content (generally called a *data hash*). A smaller fingerprint selects one or more byte sequences of varying length (also known as a *signature*). In either case, the detections are static and any changes to any byte sequence would not be a match.

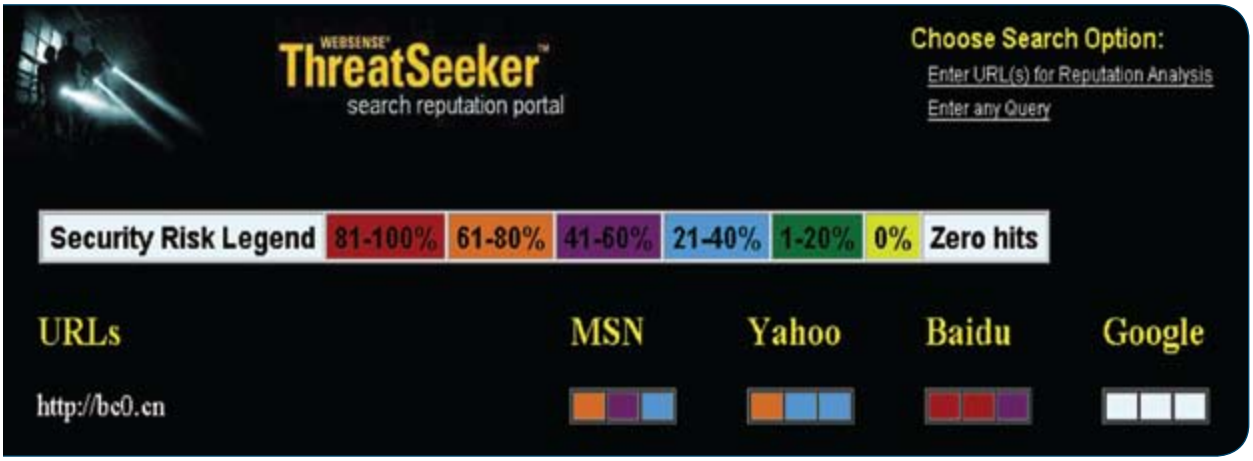
Traditional fingerprints, while large and brittle, are by far the most predominant form of classification today for Web and email threats. Most attackers today bypass them by making changes to the content. Because they are small and easily updated, fingerprinting is the classification technique of choice for reactive inbound technologies that attempt to respond and update products rapidly as new threats are discovered.

Although admittedly weak for classifying new inbound content, fingerprinting supports a clean hygiene approach to past threats. Websense products include advanced fingerprinting techniques for most inbound threats on the Web (URL lists with wildcards), email (spam fingerprints), real-time white or black lists (for third-party RWLs and RBLs) and binary applications (including licensing for third-party antivirus scanners).

Outbound content protection (also known as Data Loss Prevention, or DLP), on the other hand, advances the art of fingerprinting to identify confidential information and prevent its loss. It uses sophisticated fingerprinting techniques to create one-way mathematical representations of structured and unstructured data, which is later compared to data sent in commonly used business communications (e.g., email, Web, IM, FTP, etc.). Whereas fingerprinting for inbound threats is commonly regarded as a less effective technique, when applied to outbound threats such as data loss, it is by far the most accurate technique with less than one percent false positives. Websense Data Security Suite includes a multitude of data detection and classification techniques, including third-generation fingerprinting called PreciseID™. For more information on PreciseID™ and Websense fingerprinting, see [LID08]

4.2 Reputation

As we move from the static detection of fingerprinting to dynamic techniques, the most basic form of real-time analysis is reputation analysis. Technology that examines content reputation collects meta-information from a number of sources without actually examining the content itself. Email reputation examines the sender of the email and the IP address of the source to judge whether the email is legitimate (as in Websense True Source IP). Web reputation looks at the hostname of a Web site, the owner of the IP address, the longevity of the site or its geography. As search engines can attest, a site's popularity and the number of references to it can also aid its overall reputation score. The overriding principle for reputation filters is that knowing the history of a piece of content and its originator can tell you whether it's wanted or unwanted content. This is most valuable for classifying static content with long histories and avoiding brand new sites that lack this information on security grounds. Websense employs reputation filters and services in its products and also in more complex research analytics for reputation in Websense Security Labs to address these cases.



Although reputation is dynamic, it is not always effective. Legitimate new content sprouts up constantly, especially in today's rapidly evolving Web 2.0 world of user-generated content. This makes it difficult to establish a solid reputation, especially for high-traffic social networking sites like MySpace or Facebook. Also, more than 75 percent of all malicious content attacks today on the Web are hosted on legitimate sites that have been compromised [WEB08-B]. Consequently, everything that a reputation filter knows about a Web site can be wrong from one millisecond to the next if a site has been hacked. Finally, attackers have been known to artificially inflate reputation scores of illegitimate content as part of recent drive-by attacks such as with SEO poisoning attacks [SEC08]. As the popular saying goes, "It takes years to build a reputation but only seconds to destroy it." Reputation can help immensely in some classification domains and do spectacularly poorly in others. When reputation is not good enough to classify content from the outside, a deep analysis inside the content itself is required.

4.3 Dynamic Web Content Scanning

The holy grail of Web classification is nothing less than a complete scan of unknown content and a battery of analytics to determine its nature. Web elements—characters, words, scripts, code, tags, headers, URLs, etc.—need to be meticulously parsed and weighted to discriminate between content that is legitimate, objectionable, or malicious. The result is a classifier that identifies the nature and intent of both known and unknown Web pages in both hosted and on-premise Web security solutions.

4.3.1 Real-Time Content Categorization (RTCC)

The best general Web content classifiers (such as for Travel, Sports, and Adult Content categories) are ones that extract the most prominent elements of a Web page that are being displayed to the user and group them into specific categories. The core elements that comprise the page's identity are a mixture of primarily natural language and images but may also include colors, fonts, titles, backgrounds and other HTML artifacts.

For this task, Websense employs a machine learning model known as a support vector machine (SVM). For each category classifier, a researcher first assembles a huge collection of positive and negative examples of the category. With over 15 years of classification experience, Websense is in a unique position to be able to assemble the breadth and depth required of these sample sets from its URL database. For example, the classifier for Adult Content utilizes a collection with over one million unique positive and negative samples spanning 50 languages.

Once the collections have been assembled, a full-content parser breaks each page down into a collection of words and phrases (known collectively as terms) and counts them. It also remembers the context in which the terms are found—essentially any HTML tag that may apply to them (e.g. <title>, <script>, <img>).

The SVM is able to attach a weight to each term by examining the counts for positive and negative examples. The best terms are selected to form the dictionary for that classifier—the group of highest positive weight and negative weights for the training samples. Future pages can now be scored by a full-content parse that weighs its terms according to those that were learned in Websense Security Labs. If the page scores above a predetermined threshold it is classified to be in the target category.

4.3.2 Real-Time Security Scanning (RTSS)

The best Web security classifiers (such as for Phishing, Hacking, and Malicious categories) are not fooled by what is displayed on the page. Instead, they extract the active elements of content that define what the page is trying to do. Active elements are any kinds of content that can trigger unwanted activity—scripts, exploits, binary code and even images (if they contain remote execution vulnerabilities).

RTSS employs a regular expression engine that can logically join together any number of elements into an attack profile. For a simple example, if a Web page:

A) is located on a social networking site AND
B) has forms for login and password AND
C) has a logo or keywords that are trying to associate these forms with a financial institution, THEN

given that no bank hosts authentication credentials on social networking sites, the page is almost certainly trying to steal banking information and can be categorized as a Phishing site.

It is important to note that an RTSS profile is not a specific detection for a specific attack but rather generalizes to an entire class of attack (in this case, phishing attempts for financial credentials on social networking sites). RTSS currently supports hundreds of profiles for other attack classes, including attacks on instant messaging, program files, Web 2.0 applications, and a number of others.

The sophistication of RTSS extends to understanding the context of the content. For example, script that is active and executes code is parsed and classified differently than a description of the script—even if the actual text of the script command is the same. Although the content may be identical, one resides within valid script tags (and is therefore more dangerous) and one does not. Further, RTSS uses advanced parallel pattern searching algorithms to optimize performance well beyond traditional signature engines.

4.4 Dynamic Email Content Scanning

Email classification addresses a different technology and attack vector, but also requires full content scanning to be robust enough to catch new attacks. Envelope and message headers, message body, and attachments are among the primary elements for email examination. Combined with sender reputation and advanced fingerprinting techniques, over 99.5 percent of spam messages and malicious attachments are blocked by both on-premise and hosted Websense security solutions with misclassifications on fewer than one out of every one million pieces of content.

Email classifiers use heuristic rules to identify thousands of characteristics known to be associated with spam campaigns. These rules are individually weighted and incorporate Bayesian machine learning algorithms (similar to the SVMs used by RTCC in Section 4.3.1) to modify the final score. Collectively, they eliminate the vast majority of future attacks with a negligible misclassification rate. Interesting advances specific to the dynamic email content scanners include:

- Emails that are classified as malicious or spam and contain new Web URLs are immediately updated and blocked by all Websense Web Security products within minutes.
- Email text that is intentionally hidden within images is unmasked and classified with advanced optical character recognition (OCR) techniques.
- Image analysis tools can identify spam and objectionable visual content within emails.
- Over one million messages are sampled each day for accuracy and re-training of dynamic classifiers.
- The binary classifiers in Websense Hosted Email Service not only catch 99 percent of malicious code, but they produce false positive alerts on fewer than one out of every one million attachments scanned—a rate unmatched by antivirus scanners today.
- Email honeypots are configured to collect a huge quantity of targeted new samples of undesirable content by harvesting spam or malware on non-existent or deactivated email accounts on a host's domain.
- Dynamic email classifiers automatically re-train themselves by using the emails that are released from quarantine as a training signal.

4.5 Dynamic Outbound Content Scanning

The goal of a dynamic outbound content classifier appears fundamentally different from inbound classifiers. Rather than blocking unknown, unwanted content from entering an organization, a classifier must block known, highly sensitive content from leaving. However, many of the techniques used by Websense Data Security Suite to prevent data leak are similar to those used by inbound classifiers. Sensitive data must be discovered by distributing the work across all nodes in an organization. The data must be identified regardless of protocols, encryption, or other data transformations. The unmasked content must also be classified to determine if the type of information being sent violates a regulatory compliance initiative or if the content is sufficiently similar to confidential data to prevent its release. Fingerprinting, natural language processing, lexical analysis, and a host of other techniques known collectively as PreciseID technology are applied to this end. Finally, an understanding of the underlying context of the data transaction (e.g. reputation, identity of sender or recipient) is used to enforce appropriate policy.

Dr. Lidror Troyansky, distinguished research fellow at Websense, discusses in detail the techniques used to prevent data leakage for Websense Data Security Suite in "Information Identification: Critical Requirements for Effective Data Security [LID08]."

4.6 Multi-Tiered Classification

Thoroughly examining content can be a time-consuming affair. A strategy that Websense has adopted to distribute the computation is to employ a multi-tiered approach to classification. Any classification that can be done rapidly (such as the aforementioned fingerprints, RTCC and RTSS) will be run at client or network endpoints. When deeper analysis is required, Websense products will initiate a probe request to Websense analytics. Once Websense receives a probe request, a number of new analytics are brought to bear. The inner workings of probes are discussed in more detail in the context of the Internet HoneyGrid [WEB08-A].

4.6.1 Image Analysis

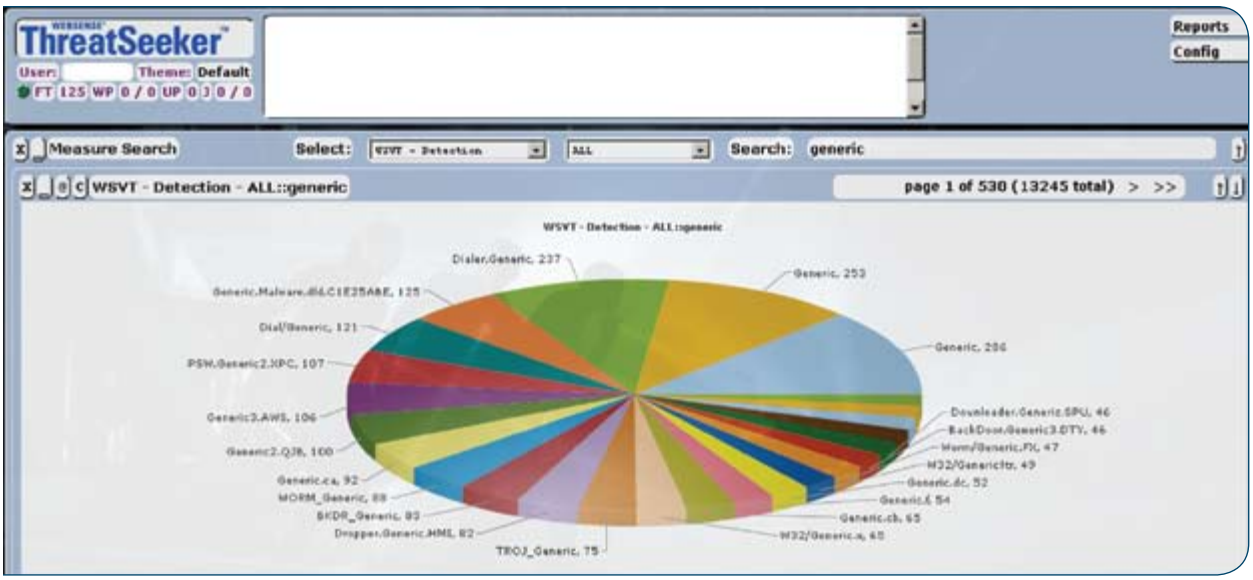
Classifying images requires a great deal of computation. Each pixel in an image (often numbering in the millions) must be fed into a series of intensive calculations to reduce it to its principal components (generally borders of shade or shape). They are then fed into a classifier that compares the principal components against known examples to identify objectionable content.

4.6.2 Link Analysis

Following the links on an email or Web page can lead to some interesting observations. Certain ecosystems of content will generally only refer to other members of the ecosystem. For example, news Web sites may mention pornography, but will almost never provide a clickable link to it. Pornographic Web sites, however, frequently cross-reference each other. Solid category distinctions begin to emerge (especially for objectionable content) after analyzing the network of links that pages use to refer to each other. The expense of downloading the thousands of links to conduct this analysis makes it a better fit as a second-tier classification technique. Websense reputation analysis is frequently collected in a similar fashion before being promoted to customer's endpoints.

4.6.3 Application Virtualization

The best way to analyze a new application is to run it and watch what happens, but that's generally not a good idea on endpoint machines. In addition to the time it takes to run such an analysis, it would not be appropriate to launch suspicious or unknown code on arbitrary systems. Consequently, Websense employs a virtualization network to launch, monitor, and classify unknown applications. All file system changes, system calls, memory changes and network calls are logged and assembled to form a profile of the executable's activity. Any program that propagates, calls back to malicious sites, installs a keylogger, etc., is automatically classified and propagated back to all endpoints in the ThreatSeeker Network. Because attackers are aware of threat virtualization networks and typically try to bypass them, a great deal of innovation arises from embedding a robust environment that fools would-be attack code into thinking it is being hosted on a legitimate system.



4.7 Classification Challenge #1: NotYourSpace

Imagine you are browsing to a friend's site on MySpace and notice a new video. You watch it. Unbeknownst to you, your MySpace top menu bar has just been replaced with an identical-looking one where all the links point to a phishing site that harvests MySpace credentials from unsuspecting visitors. Without your knowledge, your MySpace profile has been infected with the video and you've unwillingly sent several instant messages to random MySpace users to visit a pornography site that silently installs unwanted adware on unsuspecting visitor's local machines. What happened and how can you stop it?

Software vulnerabilities—small holes in a program's security model that allow an attacker to launch their own code or gain unauthorized privileges—are often at the heart of security problems today. This attack on MySpace is no exception. It is a combination of two flaws—one in MySpace that allows foreign scripts to modify the menus, and another in Apple QuickTime that allows foreign URLs to be inserted and launched when a specially crafted video is played. Just clicking on an infected video launches a hidden URL with a script that makes all of the unsavory changes to your MySpace account and further propagates the attack.

How can you stop it? Let's examine the question by process of elimination. Traditional URL filtering would fail as there would be far too many different locations being attacked almost simultaneously to keep up. Antivirus would fail as this is a Web-based attack without any application files to be scanned. Traditional firewalls would allow the traffic—it is arriving on port 80 (HTTP), after all, and you can't just turn off the Web. Reputation would not work, either. Both the source and target MySpace pages in general are already difficult to score because they present user-generated, highly-dynamic content. Further, most infected MySpace pages would have trustworthy histories with unblemished reputations. The best remaining alternative is to apply a real-time security scanner to scan the full content stream, identify the content types, and classify the malicious content on the page (QuickTime video, modified menu, malicious URLs). All access to the infected objects would be blocked. Another more draconian alternative is to turn off access to Web 2.0 and social networking sites altogether. It is likely, however, that access to increasingly mainstream Web infrastructure will increasingly become a requirement over time in the same way that other technologies like instant messaging and peer-to-peer computing have in recent times.

This worm is a real-world case, too. It spread rapidly across MySpace in December 2006 [WEB08-D] and represents a classic playbook (cross-site scripting + vulnerabilities) attack being used by attackers to infiltrate social networking sites today.

4.8 Classification Challenge #2: Save or Spend?

Imagine being the chief security officer at a large retail chain and having a critical requirement to protect customer credit card data from being stolen. You decide to block all outbound traffic for groups of numbers that look like credit card numbers. As soon as you turn on the new policy, alerts begin popping onto the dashboard in alarming numbers. As a disproportionately large group of your team begins to frantically examine the alerts, you observe that the data leaving the network is not credit card data. In fact, it is coupon data—information that is disseminated to over 1,000 organizations on a regular basis to collect reimbursement from customer-submitted coupons. Unfortunately, the size of the data and the formatting match your credit card filters. What can you do to enable your business by allowing the coupon data transmission while simultaneously preventing credit card data from leaking?

The general perception is that protecting essential information protection on outbound data is much easier than on inbound data because there are far fewer unknowns; an organization should presumably know exactly what kind of data it wants to protect and be able to shape policy to prevent its external transmission. As this real-world issue solved for several Websense Data Security Suite customers demonstrates, classifying outbound data can be just as difficult. An outbound data classifier must learn the context surrounding the proprietary information in addition to the raw data. This can include the format of the data, the content and language that surrounds it, the source and destination of the data, the protocol used to transmit the data, and a host of other characteristics.

5 Adaptation

The process of discovery, identification, and classification is a comprehensive strategy for analyzing content, but it has one major flaw: it is tuned to a specific instant in time. What happens in the next moment, hour, week, month or year? Content, the technology that drives it, and the way in which it is accessed and used changes all the time. Yesterday's classifiers and their data are tomorrow's wasted CPU cycles and disk space (although it is surprising how much third-party testing remains dedicated to boot sector and DOS virus scanning these days). Given the incredible velocity of change in the computing landscape today, how can a classification system keep up? Websense follows the two fundamental adaptive principles of *visibility* and *realignment*.

5.1 Visibility

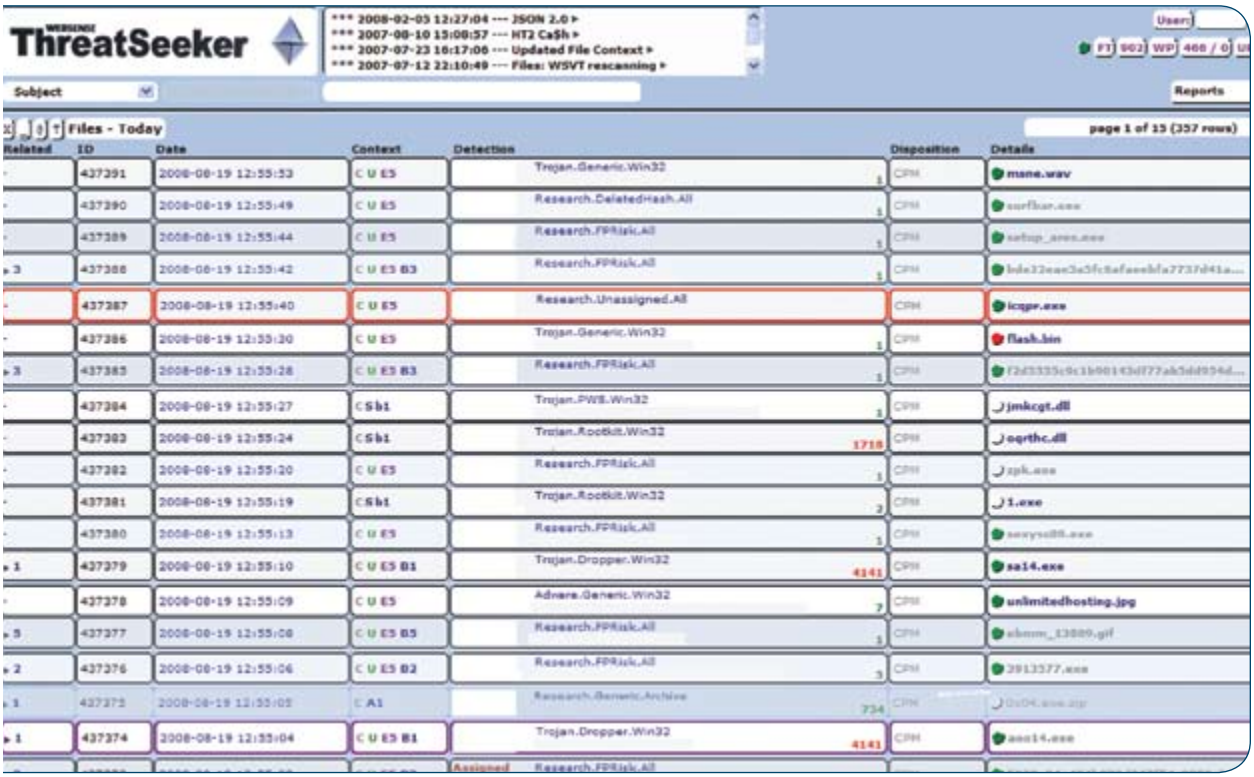
The first principle of adaptation, *visibility*, is simple in theory but difficult in practice. In order to determine how to evolve a classification system, you must monitor 1) how the content itself is evolving in usage and 2) how well the current classification system is performing. In other words, you only need to understand how all the data on the Internet and inside the network is being used and make sure you're accurately classifying it all across the globe. Trivial, right? Hardly.

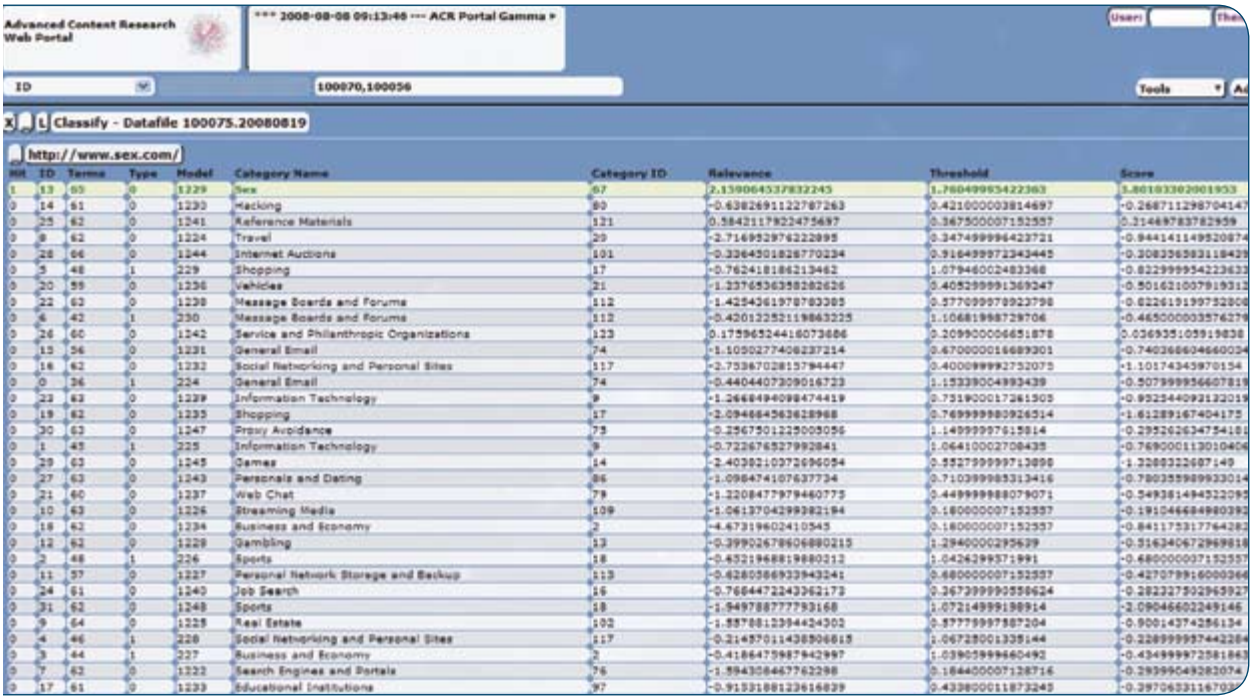
5.1.1 ThreatSeeker Network

By monitoring and probing the data usage of more than 50 million systems in real-time, the ThreatSeeker Network is the engine that drives the necessary data collection for large scale visibility [WEB08-A]. Statistical data from over one billion pieces of content is collected every day, spanning geography, content type, content category, and hundreds of custom probes targeting data of current interest. Many of these probes are fuzzy in that they are lax classification profiles that cast a wide net over broad phenomena and don't have to be exact matches like fingerprints. Trend analysis will indicate spikes in content types, locations, or specific profiles that require more examination. The probe data will be studied and will eventually turn into targeted and highly-specific analytics and classifiers.

5.1.2 Tracker

Websense examines the mountain of data mined by the ThreatSeeker Network by the Tracker, a huge database of cross-linked information from all sources. Everything that Websense knows about content is stored in the Tracker—the data itself (URL, binary, email, and other instances, deeply parsed in a manner appropriate to the content type), as well as everything that is known about the data (e.g. geography, how it is linked to other data, history, results from all analytics). The Tracker provides a Web-based interface with dozens of panels that allow researchers to easily group the relevant pieces of data together. Most data mining research is a combination of the Tracker's data visualization capabilities with a series of well-chosen database queries.





5.2 Visibility Realignment

Once a new trend becomes visible, the classification system realigns its data collection to collect the target information at its source. For incremental changes, the realignment process is a combination of automatic adjustment in real-time to collect and classify the content more robustly. For brand new data classes, the research team must build and embed the necessary infrastructure into the classification system.

5.2.1 Discovery Realignment: ThreatSeeker Data Miners

An important part of the data collection process includes the ThreatSeeker Data Miners. These systems assemble the collective data from the ThreatSeeker Network and search more deeply through the most interesting phenomena. Areas of interest always include security outbreaks (both within and across domains), but may also include a sharp spike in general traffic to a region or regular checking of high-traffic domains known to be associated with security risks.

5.2.2 Identification Realignment: Content Type Update

If a new protocol, file format, unpacking tool, etc., is discovered in mainstream use, new analyzers are created and placed at all tiers of the Websense ThreatSeeker Network. Having made the necessary transformations on the new data class, the classification process can now proceed as it does with all other content.

5.2.3 Classification Realignment: Automatic Retraining

An innovation for automatically updating classifiers is the method Websense uses for incrementally retraining classifiers based on the latest incoming data. As mentioned in Section 4.3.1 and 4.3.3, dynamic content classifiers use large numbers of examples to learn and then classify future cases. Any errors submitted by customers (either through sample submission for Web content or quarantine release for email content) are fed back to the list of examples. First, these samples are added to the correct collection for future training. Second, statistically similar samples that are in the wrong collection are also removed. Classifiers are constantly retrained with the new sample sets and automatically improve to progressively higher detection rates and lower false positive rates.

5.2.4 Classification Realignment: Good ol' R&D

The quality of a classification system is defined by the team of researchers and engineers behind it. Applying solid research principles to identify and classify a content class to a high degree of accuracy is a tremendous challenge in itself, especially in the realm of security. That cost does not include the foundational team to support and enable the research—code developers for the product-side analytics, Web developers for the data visualization, infrastructure developers to support a tremendous array of systems processing volumes of data, and general maintenance to keep the whole system operational.

5.3 Adaptation Challenge: The Case of the Vanishing HTML

Imagine that you visit a Web page and are especially impressed with the design. It contains the works—drop-down menus, interactive maps, animated graphics and text flying across the screen—and it's all clickable. Curious about how such fancy Web pages are created, you decide to view the HTML source. To your dismay, there's only one line in it—a reference to an object with a .SWF extension. You realize that the content on this Web page is entirely contained in a single Shockwave file, and there will be no HTML text or context to examine. You wonder if this Web page style is common and how a real-time classifier would have any chance to classify it.

After observing just a single case of the phenomenon, a curious researcher in the Websense Security Labs inserts a visibility probe into the Internet HoneyGrid to detect instances in real-time of Web pages that contain little more than links to Shockwave files. After a few days and a few billion page views, the statistical data indeed validates that this is a substantial trend. The researcher adapts the ThreatSeeker Network by implementing a new identification component to parse the SWF format, store the essential information in the Tracker, and feed the purified content back into existing classifiers for automated analysis. Now when the probe discovers new uncategorized Shockwave pages, they are now correctly parsed for their true content, classified and automatically updated every few minutes.

Examples abound of this phenomenon [FLA04] and numerous other micro domains that are classified by Websense solutions. Adding fingerprints for each example in each of these domains would not scale. The only successful formula to generalize classification across huge ranges of content is to discover, identify, classify, adapt, and repeat.

6 Conclusion

The problem of classifying all network content is an ambitious one to tackle. The creation of the technologies that comprise the Websense Content Research Cortex is an investment of more than 15 years of classification experience. Websense creates and utilizes a vast array of classification techniques to block malicious or unwanted data from entering the network and protect confidential or proprietary data from leaving. Now with more than 100 researchers working worldwide every day, the task evolves as the Internet evolves in a continuous cycle of discovery, identification, classification, and adaptation. Essential Information Protection requires nothing less.

About Websense

Websense, Inc. (NASDAQ: WBSN), a global leader in integrated Web, messaging and data protection technologies, provides Essential Information Protection for more than 42 million employees at more than 50,000 organizations worldwide. Distributed through its global network of channel partners, Websense software and hosted security solutions help organizations block malicious code prevent the loss of confidential information and enforce Internet use and security policies.

Websense Security Labs

Websense Security Labs is the security research arm of Websense, Inc. that discovers, investigates and reports on advanced Internet threats. Unlike other research labs, Websense has an unparalleled knowledge of malware and where it resides on the Web. This allows Websense solutions to detect and block new threats that traditional security research methods miss, enabling organizations to protect sensitive content from theft, compromise, or inappropriate use. Recognized as a world leader in security research, Websense Security Labs publishes findings to hundreds of security partners, vendors and other organizations around the world and provides security metrics to the Anti-Phishing Working Group.

Websense Security Labs – a Pioneer in Emerging Threat Protection

- Unparalleled visibility and discovery on a massive scale
- Real-time adaptive ability to respond to trends and threats in a Web 2.0 world
- Powered by a unified world-class research team
- Many first discoveries, including the unpatched, high-risk Microsoft Excel vulnerability (March 2008)
- First to market with phishing protection
- First to market with drive-by and backchannel spyware protection
- First to market with bot network protection
- First to market with crimeware and keylogger protection

Security Alerts

Register with Websense Security Labs to receive FREE security alerts about malicious Internet events, including spyware, spam, phishing, pharming, and corrupted Web sites.
http://www.Websense.com/securitylabs/alerts/

Blog Highlights

The Websense Security Labs Blog delivers the most current information and breaking news about security research topics and today's advanced Internet threats. Websense Security Labs investigates and publishes information about outbreaks, new threats, and other relevant Web security topics to protect organizations from increasingly dangerous Internet threats. For more information, check out our blog: http://www.Websense.com/securitylabs/blog.

7 References

[WEB08-A] Websense, "The Websense ThreatSeeker Network: Leveraging Websense HoneyGrid Computing," http://www.Websense.com/Docs/WhitePapers/WP_HoneyGrid_Computing.pdf, 2008.

[WIK08-A] Wikipedia, "Protocol (computing)," http://en.wikipedia.org/wiki/Protocol_%28computing%29.

[WIK08-B] Wikipedia, "Encryption," http://en.wikipedia.org/wiki/Encryption.

[MCR08] Joren McReynolds, "Packer Detection and Generic Unpacking Techniques," http://www.Websense.com/securitylabs/blog/blog.php?BlogID=176, February 29, 2008.

[BRU08] Nicolas Brulez, "Unscrambling Custom Obfuscation and Executable "Infection," http://www.Websense.com/securitylabs/blog/blog.php?BlogID=178, March 12, 2008.

[CHE07] Stephan Chenette & Alex Rice, "Automatic JavaScript Deobfuscation," http://www.Websense.com/securitylabs/images/alerts/wsl_pacsec2007_en.pdf, PacSec 2007, November 30, 2007.

[HUB07] Dan Hubbard, "HoneyJax (AKA Web Security Monitoring and Intelligence 2.0," http://www.Websense.com/securitylabs/images/alerts/honeyjax_defcon2007.pdf, DEFCON 2007, August 5, 2007.

[WEB08-B] Websense, "State of Internet Security," http://www.Websense.com/securitylabs/docs/WSL_Report_1H08_FINAL.pdf, July 2008.

[SEC08] SecurityFocus, "SEO poisoning attacks growing," http://www.securityfocus.com/brief/701, March 12, 2008.

[BOY07] Chris Boyd, "Skype Worm Variant Targets Other Instant Messaging Clients," http://blog.spywareguide.com/2007/05/new_skype_worm_variant.html, May 23, 2007.

[WEB08-C] Websense, "Mass Attack JavaScript Injection," http://securitylabs.Websense.com/content/Alerts/3070.aspx, April 22, 2008.

[LID08] Lidror Troyansky, "Information Identification: Critical Requirements for Effective Data Security," http://www.Websense.com/docs/WhitePapers/PA_Information_Identification_Fingerprinting.pdf, 2008.

[FLA04] FlashJournalism, "Tour De France 2004 - Publicité," http://flashjournalism.com/examples/en/index.html, 2004.

[MIC08-A] Microsoft, "Vulnerabilities in Microsoft Word Could Allow Remote Execution (951207)," http://www.microsoft.com/technet/security/Bulletin/MS08-026.mspx, May 13, 2008.

[MIC08-B] Microsoft, "Vulnerability in Microsoft Publisher Could Allow Remote Execution (951208)," http://www.microsoft.com/technet/security/Bulletin/MS08-027.mspx, May 13, 2008.

Appendix A – Partial List of Supported Websense Protocols

Instant Messaging/ Chat	Google Video	Netease Popo	ssh
AOL Instant Messenger or ICQ Attachments	Google Web Accelerator	NetMeeting	TeamViewer
AOL Radio	Gopher	NFS	Telnet
Ares (prior to v1.8.1)	GoToMyPC	NNTP	Tencent QQ (see the Knowledge Base)
BeInSync	Hamachi	NTP	Terminal Services
BitTorrent	Hopster	Onshare	TongTongTong
Bot Networks	Hotline Connect	OpenWindows	Toonel
BoxCloud	HTTP	Paltalk	Tor
Brosix	HTTPS	Pando	TryFast Messenger
Camfrog	ident	pcANYWHERE	VNC
Citrix	IMAP	pcTELECOMMUTE	Vyew
ClubBox	IMVU	PeerCast	VZOchat
Damaka	IRC	POP3	WAIS
daytime	iTunes	pptp	WallCooler VPN
DirectConnect	JAP	Project Neon	Wavago
eDonkey	JetCast	Qnext	WebEx (PCNow & Support Center)
Email Borne Worms	LDAP	Raketu	Wengo
Eyeball Chat	Liquid Audio	RealTunnel	Windows Media
EZPeer	LogMeIn	RTSP (QuickTime RealPlayer)	WinMX (prior to v3.31)
FastTrack (Kazaa iMesh)	Lotus Notes	SHOUTcast	Woize
Finetune	Meetro	SIMP (Jabber)	Xfire
finger	Metacafe	Skype	X-IM
FolderShare	Microsoft HTTPMail	Slingbox	Yahoo! Mail Chat
FTP	MindSpring	SMTP	Yahoo! Messenger
Gadu-Gadu	MSC Messenger	Social FM	Yahoo! Messenger Attachments
GhostSurf	MSN Messenger	SOCKS	Your Freedom
GigaTribe	MSN Messenger Attachments	SocksOnline	YouTube
Gizmo Project	MyIVO	SoftEther PacketiX	Zolved
Gmail Chat	MySpaceIM	SoonR	
Gnutella (Morpheus Xolox)	NateOn	SoulSeek	
Google Talk	Neos	SQL Net	

Appendix B – Partial List of Supported Websense Languages

Albanian	French	Polish
Arabic	German	Portuguese (Brazil)
Basque	Greek	Portuguese (Portugal)
Belarusian	Hebrew	Romanian
Bulgarian	Hindi	Russian
Catalan	Hungarian	Serbo-Croatian
Chinese (Simplified)	Icelandic	Slovak
Chinese (Traditional)	Indic	Slovenian
Czech	Indonesian	Somali
Danish	Italian	Spanish
Dutch	Japanese	Swedish
English	Kazakh	Thai
Esperanto	Korean	Turkish
Estonian	Kurdish	Ukrainian
Ethiopian	Latvian	Urdu
Farsi	Lithuanian	Vietnamese
Finnish	Norwegian	