TRUSTEDKNIGHT™

PROTECTOR **AIR**

Exploring Full Transaction Stack Protection through a single, simple-to-deploy solution.

Overview

For most organizations web applications are not just the most visible part of the business, but also a critical method for customers to access private information and engage in sensitive transactions. As such, organizations should ensure that they provide a safe, reliable, and secure environment for their customers. However, in practice many businesses fail to consider the full scope of the application environment when evaluating threats and security measures, or they take a piecemeal approach, combining multiple solutions with each only having limited visibility into the application environment.

Instead, organizations should consider all aspects of the web application threat environment. Extending a web application's security strategy to the website, the end-user, and the web sessions themselves, as well as integrating security with fraud intelligence is an approach Trusted Knight calls Full Transaction Stack Protection. Trusted Knight offers Full Transaction Stack Protection through a single, simple-to-deploy solution: Protector Air.

Web Application Threat Environment

PROTECTING THE WEBSITE

Traditional web security strategies focused entirely on protecting the web server itself from attack. This is reasonable since not only is it the most accessible target – it is by design publicly accessible on the Internet and at least partly open to all visitors – it is also the richest target. The web server is the hub through which all online interactions flow. An attacker who targets a single user has access only to that user's information, but an attacker who targets a website can potentially have access to information on all users.

Application-level attacks have driven the development of Web Application Firewalls and related solutions that seek to filter out threatening web traffic before it reaches the website. This essentially creates a shield around the website, protecting the hub of all web transactions.

Yet this server-only approach is incomplete. Any transaction involves multiple parties, and web application transactions are no different: the website is one participant, and the end-user, or customer, the other. For any single transaction both participants are equally involved and both need to be included in any risk analysis and security approach.

CONSIDER THE RISK TO THE USER

There are many advantages to an attacker targeting individual end-users instead of the website. End-users, especially consumers, are usually much softer targets. They typically have fewer defenses – at best simply using a traditional signature-based antivirus solution (which is more than likely not up-to-date), at worst running no additional security software. Their computer is also much more likely to be behind in applying software patches to address vulnerabilities than the server running the web application. In addition, users visit a wide range of sites, most are non-business-related, so are much more likely to click a link, get fooled by a pop-up or phishing email, or visit a website that distributes malware.

Attacks against users are also much more likely to go undetected, even if the volume of compromised user devices is high. This is in part because with one malware campaign an attacker can actually attack many targets. Consider a banking trojan that exfiltrates credentials from users visiting banking sites – since each user would likely only bank with one or two different banks, the attack is spread among dozens of banks (with each bank receiving a smaller subset of the attack volume), rather than concentrated on a single bank.

This is analogous to the shift in approach of network security. Just as the old notions of protecting the perimeter with network firewalls had to evolve in the face of remote user access, distributed applications, and business-to-business integration, the application "perimeter" needs to be extended to include the users' end of the web session. Expecting the user to be responsible for his own security and blaming him if his device is compromised is unrealistic and ultimately results in customer ill-will, increased support costs, and, in some industries, liability lawsuits.

Once an organization starts to include both sides of a web application session in-scope for security, it becomes clear that using separate, unintegrated security solutions is only moderately effective. There are some attacks that just will not be caught unless both the user-side and the web server-side are evaluated together, as one web session. There is also a lot to gain with this more comprehensive picture, which can give organizations an edge against fraudulent transactions as well.

For example, consider the approach most endpoint protection products such as traditional antivirus take. The focus of these products is on stopping and cleaning up malware infections on a computer. While this is a respectable goal, it can miss the point. For one thing, almost three decades of antivirus development has demonstrated that this is not a war that can be definitively won, but an ongoing struggle with both sides innovating to counter the other. So end-user computers do get infected by malware, and that malware runs for a period of time before the antivirus software recognizes it, and cleans or quarantines it.

What about MFA?

Sometimes multi-factor authentication (MFA) is implemented on critical web applications to help mitigate the risk of credential theft. The idea is that simply using a stolen username and password will not be enough to log in if something else is required. While this can be an effective layer in a security approach, it suffers from a few limitations.

The first is simply that not all sites can afford the disruption of MFA. It may work for a financial account where users are more willing to accept the additional steps. But for an ecommerce site where users can login to speed up checkout, users are more likely to consider it too inconvenient and shop elsewhere.

Where MFA can be used, the level of security highly depends upon the method used. Some websites rely on SMS messages, since this is inexpensive to implement and support, and is something with which most users will be familiar. However, SMS is known to be weak. It is vulnerable to interception either by mobile phone malware or through compromising the underlying 1970s-era Signaling System 7 (SS7) which routes messages among providers. On a less technical level, a criminal can employ social engineering to trick the user's mobile phone company into reassigning the number to a phone the attacker controls.

Even more secure MFA techniques can only be partially effective at stopping fraud. Clever malware running on the user's device can simply lie in wait until the next time the user accesses the site and completes the authentication process. At that point it can access the user's account information, manipulate transactions the user initiates, or execute side transactions without the user's knowledge.

This driving focus on keeping the computer clean raises some concerns. If there was a period of time when the malware was active on the user's computer, what was it doing? In many cases once the antivirus vendor's research team identifies how to recognize the malware and clean the computer, they move on to the next threat. The analysis of what harm is caused extends only to the technical impact on a computer. They may say it contains a keylogger, for example, but will not track or know what data was stolen. Was the keylogger monitoring specific websites? Should the user change their passwords on certain websites? Or notify these websites that their accounts may be compromised? This type of malware can also alter a website's pages as viewed by the user, and even manipulate transactions. Was additional data stolen? Should the user review transactions on certain websites? What harm is actually inflicted on the user (not just the computer)? Cleaning the malware from the computer does not clear things up for a user if the malware has already exfiltrated login credentials, stolen personal information, or executed fraudulent transactions on the user's account.

It is only by having a security strategy that includes both the web server-side and the user-side of the transaction, and is aware of and mitigating the security threats to both that this problem can be effectively addressed.

FRAUD MONITORING AS A SECURITY STRATEGY

Another example of not considering the entire scope of a web application environment is in the use of fraud-monitoring solutions. While they can be an effective component of a security and anti-fraud strategy, it is sometimes the case that an organization will choose to focus exclusively on detecting fraud. This may be especially tempting in financial services and payments, where fraud losses can have a high business impact, outweighing all other concerns. The reasoning is that if an organization can monitor all transactions and either actively block or flag for review any anomalous ones, then that should mitigate fraud. An assertion given in support of this may be that upon putting such a solution in place, fraud does indeed drop, therefore that's all that is required. However, just as focusing on web security limits effectiveness, so too does this approach.

The fact is that almost all fraud is preceded by a security incident. A criminal hacks in to an organization and submits bogus money transfer requests AFTER attacking the web application engine to exploit unpatched vulnerabilities. A criminal initiates withdrawals to drain a user's bank account AFTER installing malware on the user's computer and manipulating an otherwise-valid web transaction. A criminal impersonates a legitimate user with valid credentials AFTER repeatedly trying the login page with a large set of stolen credentials to find a valid one.

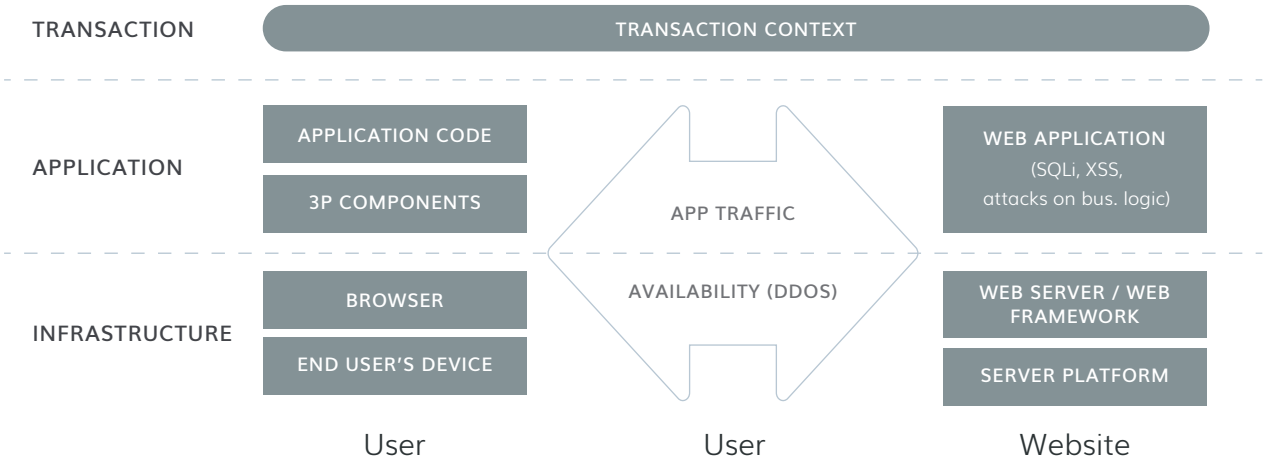
In a very real sense, focusing on fraud-monitoring and neglecting security is ceding most of the battlefield to the criminals. The reality is that they work best together, building on each other. Effective security can in most cases prevent the fraud from happening in the first place, or at least provide an early indicator of future fraud. And fraud-monitoring can provide information that can be used to tighten security controls and sharpen defenses.

Here is one example: consider a user whose computer has been infected with malware, a banking Trojan. This malware doesn't just exfiltrate credentials, but actually piggybacks on a valid, authenticated session initiated by the user. Running in the same context, with the same device fingerprint and same session ID as other valid transactions, this malware modifies one of the requests the user initiates such that money is transferred to the attacker's account. Detecting this by using fraud-monitoring alone would be challenging. But with the proper security approach the attack could either be blocked outright, or the detected security event could provide intelligence which the fraud-monitoring system can include in its risk analysis.

Full Transaction Stack Protection

Any organization that has a web application that deals with sensitive information or is used for transactions needs to include all aspects of the web threat environment into the security and anti-fraud strategy. Trusted Knight calls this "Full Transaction Stack Protection," and it includes:

1. Protecting the website from direct attack on the infrastructure, frameworks, and application logic
2. Protecting the user from endpoint and browser-based malware such as keyloggers and banking Trojans
3. Protecting the communications from service disruption through distributed denial-of-service (DDoS) attacks
4. Incorporating anti-fraud intelligence into the defense to protect transactions



Expanding an organization's security strategy to include all participants – internal and external – and all layers – infrastructure and application – is the only way to truly address threats to the web application. Furthermore, coordinating the security strategy with anti-fraud monitoring can provide full transaction stack protection, dramatically reducing technical risk as well as business fraud. This section examines each aspect of the transaction stack along with the risks and common attacks.

PROTECTING THE WEBSITE STACK

Internet-accessible websites are almost constantly being scanned and probed for vulnerabilities. In most cases these are automated, performed by bots that are looking for possible attack targets, or by worms and other malware seeking to spread to other vulnerable servers. Such scans are also done by attackers who have already targeted an organization and are looking for an entry.

The Website Infrastructure

Vulnerabilities in the infrastructure used by a website are the most commonly probed. This includes:

- The underlying operating system on which the web server is running
- The web server itself, usually Apache or IIS
- Any web frameworks or UI frameworks on which the application code is built

Any new vulnerabilities against infrastructure components are typically widely circulated, and organizations who fail to follow regular system patching and hardening practices have higher exposure. One high-profile example of this was an Apache Struts vulnerability that went unpatched for months at Equifax, leading to the theft of highly sensitive information on over 140 million people in 2017.

The Web Application

More sophisticated attacks will target application-level vulnerabilities. Many of these are general techniques, such as exploiting poor validation in applications for SQL Injection, file inclusion, and cross-site scripting. These are the most common – SQL Injection alone typically accounts for roughly half of all application-level attacks[1] – because they do not require special application knowledge and because so many websites are susceptible. These and other attacks are ranked among the Top Ten web application security risks published by the Open Web Application Security Project (OWASP) precisely because they are more difficult to mitigate and require web development teams to continually check for in web application code.

But these higher-level attacks can also be very application-specific, exploiting flaws in the business application logic itself such as improper session management, privilege escalation, poor error handling, information leakage, etc. Because every website is different, a determined attacker with a specific agenda can almost always find high or critical vulnerabilities in a target's website.

Web application-level attacks can have a very high impact, as they attempt to reach through the web server front end and attack the application logic or backend databases, potentially accessing customer account data or personal information, altering or falsifying data, and manipulating transactions for theft or fraud. These can also be used to gain entry for deeper attacks within a corporate network.

[1]Akamai Q4 State of Internet Security Report

How vulnerable are Web Applications?

30%

In 2017, 30% of reported breaches involved attacks on web applications...

93%

...and of those, 93% of these were financially motivated or perpetrated by organized criminal groups.

Verizon 2017 Breach report

50%

Close to 50% of applications remain vulnerable – meaning they have an unpatched "critical" or "high risk" vulnerability - on every single day of the year.

FINANCIAL SERVICES	44%
RETAIL	59%
ACCOMMODATION & FOOD SVCS	59%
HEALTHCARE & SOCIAL ASSISTANCE	5%
PUBLIC ADMINISTRATION	52%

WhiteHat Security 2017 report

PROTECTING THE COMMUNICATIONS

In addition to direct attacks on a website, there are attackers who can cause harm by interfering with the traffic that flows between the users and the website.

Attacks on infrastructure availability

It's becoming increasingly common for criminal groups (or individuals) to target websites with distributed denial-of-service (DDoS) attacks. These most commonly operate at the network layers (layers 3 and 4), using a botnet to overwhelm the target website with Internet traffic, often by reflecting and amplifying it through other Internet-connected servers. This high volume so overwhelms the resources of the web server or network infrastructure that the website becomes unavailable to legitimate users. This is most often done to businesses as a straightforward extortion technique: pay the attackers to forestall or stop the attack. There can be other reasons behind these attacks as well, one of the more dangerous being to use DDoS as a smokescreen to provide cover for a more targeted attack.

Attacks on the application traffic

Harder to handle are DDoS attacks that affect the web application layer (layer 7), such as using a large number of compromised computers or IoT devices to make legitimate-looking requests for website content. Since this is valid HTTP and HTTPS traffic, it cannot be dropped as simple network noise. And if it is originating from a sufficiently large number of devices it can be challenging (if not impossible) to tell which requests are truly from users vs. bots. Responding to this volume can rapidly overwhelm the web application's scalability.

PROTECTING THE USER-SIDE STACK

Attacks on the end-user can take multiple forms as well. User-side malware can take many forms including stealing private files, stealing processing power for compute-heavy tasks such as cryptocurrency mining, extorting money through ransomware, and just causing disruption for the user. But insofar as it concerns the web application, malware that can interfere with a user's web sessions, exfiltrate sensitive data, and ultimately commit fraud are of most concern.

The user's "infrastructure"

Just as the infrastructure of the web server has risk, so too does the device and platform the user is using. This includes the user's device platform, mobile or desktop, the operating system, and the user's web browser.

Attacks on user devices come from many directions, and are in practice impossible to completely prevent.

EMAIL Sending users phishing emails with malicious file attachments, or with links that cause them to download malicious files. This technique has been around almost as long as email, yet continues to be a major issue as criminals continue to evolve their evasion techniques.

WEB BROWSING Drive-by downloads, which can cause a user to download malware simply by visiting a website, clicking on a deceptive pop-up, viewing an online advertisement, etc.

TRUSTED APP STORES Delivering malicious code in browser extensions, which appear safe because they are available in a web browser's app store.

COMPROMISED SOFTWARE Embedding malicious code into the installer for a legitimate software package or mobile app, often without the knowledge of the software company.

Attacks at this level are usually not brute-force direct attacks but rely on subterfuge to trick the user into running malicious code or installing endpoint malware.

The end result of this is a user with an infected device who has no knowledge of the compromise. When focused on fraud, this malware will often lie in wait and monitor user activity, particularly websites visited, looking for interactions with sensitive applications. Commonly called banking Trojans (when targeting banking applications) or keyloggers this malware can be used for fraud in several ways.

Keylogging/Form Grabbing - The most common effect is simple keylogging: capturing the data the users enter (such as login credentials) and exfiltrting it to a command and control location (C2).

Web Page Tampering - This malware can also modify the way the web page is displayed to a user – for example, adding additional fields to a form to extract more sensitive data, or modifying links on a page.

Transaction Manipulation - More sophisticated malware payloads can actually be used to either submit fraudulent transactions directly or modify transactions as the user interacts with a site.

The application running in the user's browser

Even if the user's device is free from malware, malware can still be running while the user is interacting with the web application. This is because the web application itself may include malicious code. This is usually in the form of javascript malware. Javascript is supported by all modern web browsers – it has to be, since almost all websites use javascript in some fashion[2]. There are two main methods that can be used by an attacker to distribute malicious code in a web application.

First-party distribution - The website itself may be the source of the malicious code, most commonly because a criminal hacked into the server and modified the application to include the javascript malware.

Third-party distribution – Most modern websites pull a large portion of their content from third-party sources, including javascript libraries, tools such as analytics and tracking scripts, and plug-ins for content management systems such as WordPress. This greatly increases the attack surface since any one of these can be a vector for javascript malware.

[2]94.9% according to w3techs.com

Third-party distribution is one of the most often overlooked vectors for website malware.

A recent trend is to load cryptojacking code (cryptocurrency mining javascript) into a web page so that any user visiting the site will also run the malware and thus unwittingly contribute CPU power to the attacker's cryptomining pool. In early 2018 Text Help, a provider of a third-party assistive technology tool was hacked and one of their script files was modified to deliver such cryptocurrency mining code. As a result over 4,000 websites, many of them UK and US government sites, were serving this malware.[3]

Stealing CPU power from users may be relatively harmless, but this delivery chain can also be used for more malicious payloads. In late 2017, a similar threat to WordPress websites that appears to have started out as cryptocurrency mining malware evolved to deliver keylogging javascript that stole all user-entered data and exfiltrated it to the attacker's domain. It hid behind the appearance of legitimate analytics tools and was particularly dangerous on WordPress sites that served as ecommerce platforms, stealing personal and payment data.[4]

Popular websites may notice this and mitigate it relatively quickly, since all of their website users will be affected. But many websites that are less-trafficked or poorly maintained will continue to serve this content – in the case of the WordPress keylogger, as of January 2018 there are still over 2000 websites serving the malware.[5]

[3]https://techcrunch.com/2018/02/12/ico-snafu/

[4]https://www.scmagazine.com/wordpress-hit-with-keylogger-5400-sites-infected/article/712733/

[5]https://thehackernews.com/2018/01/wordpress-keylogger.html

While not as dangerous to the user as endpoint malware in general, javascript malware can be just as damaging to the user's web session with a specific web application. This includes keylogging, page tampering, and even transaction manipulation.

PROTECTING THE TRANSACTION LAYER

The above sections discussed what are essentially security risks to the user, the website, and the communications. Successful attacks against these targets are most often what lead to fraud, and mitigating these risks can effectively reduce fraud. However, without being transaction-aware and specifically looking for fraud, security defenses alone cannot eliminate fraud.

Some of the attacks that can occur at this level are:

Impersonation - an attacker using stolen credentials (or other sensitive information such as a credit card number and associated details) to impersonate a legitimate user and initiate unauthorized transactions. This can be especially hard to detect if the attacker has also stolen the device fingerprint, such as OS version, web browser, etc. and uses this as part of the impersonation.

Credential Stuffing - using a database of stolen credentials from another source and trying them in succession to see if any of the accounts exist with the same password on the target website (and then impersonating these users)

Covert Session Hijacking - malware on the user's computer piggybacks on an existing user session (i.e. after the user has successfully authenticated) to privately access web content or to initiate additional unauthorized transactions as the logged-in user.

Transaction Hijacking - A specific case of session hijacking where instead of covertly initiating additional requests to the website, the malware actively manipulates user-initiated transactions (such as payment amounts or recipients) as they occur.

Many of these attacks cannot be effectively stopped without being aware of the context of the transaction, as well as the participating user and web application.

Protector Air: Full Transaction Stack Protection

by Trusted Knight

Trusted Knight's Protector Air is a simple, comprehensive security solution providing full transaction stack protection for business-critical web applications. It is designed to protect against all of the threats highlighted above that result in security breaches and fraud.

SIMPLE TO ACTIVATE FOR ANY WEBSITE

Protector Air is specifically designed to minimize impact on the website. All that is required is a DNS change to redirect all web traffic through Protector Air and an SSL certificate to secure the connections.

Zero Deployment Cost

Because Protector Air is a cloud service, there is no hardware or software to provision and no software to setup, configure, or maintain.

Platform Independent

Protector Air does not require any modification to the website or application, and can work with websites running on any platform. The customer does not need to add any javascript code or script snippets to web pages, or integrate with any API, or modify any application logic to perform fraud checks. Because Protector Air runs in-stream, it is able to provide all of this functionality on the fly, with no impact to the website.

Deployment Independent

Because of this zero-impact to the web application, Protector Air is ideal for any website deployment architecture: self-hosted in an organization's datacenter, hosted in the cloud or with a hosting provider, even for websites completely outsourced to a service provider.

Highly-Available

Protector Air runs within the Amazon Web Services worldwide infrastructure, taking full advantage of AWS's high performance, scalability, redundancy, and global availability. The service automatically scales to handle spikes in traffic volume, and is maintained in a highly-available infrastructure across multiple availability zones (i.e. separate datacenters). In addition, Protector Air builds upon the core AWS Shield service and is highly resilient to DDoS attacks.

Agentless, Invisible Protection for End-Users

Similarly, Protector Air is designed to be frictionless for the end-user, having zero impact on the user experience. There is no end-user software required to provide a level of protection for all end-users visiting the website.

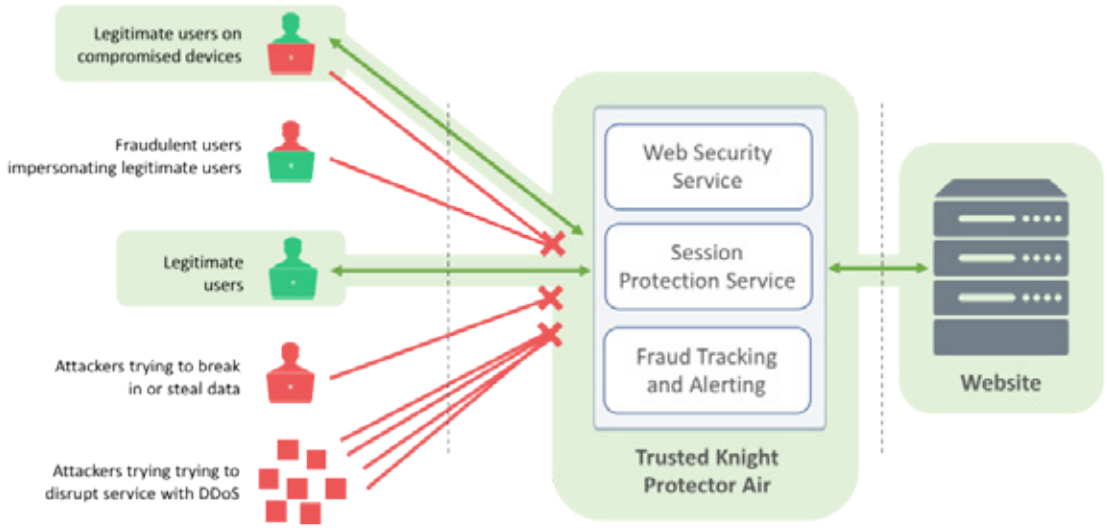
Supports All Platforms and Browsers

Protector Air uses standard web technology and javascript and is designed to work in all modern web browsers. Furthermore, while the threats and risks are different on mobile devices and desktop operating systems, Protector Air's protection is cross-platform. Because Protector Air works by layering protection onto web sessions without altering the user experience, there is no need to maintain a diverse set of versions, instructions, and support for users with separate iOS and Android Apps, or different plug-ins for PCs vs. Mac, etc.

How Protector Air Works

Protector Air is a cloud-based service that sits between the end-user and the website, proxying all interactions and extending security and anti-fraud protection to both. It monitors all web application traffic between the website and the various computers, mobile devices, and other endpoints trying to interact with the website.

- Legitimate users, who need safe, reliable, and secure access to the web application
- Legitimate users on compromised devices, where malware is trying to steal information or manipulate sensitive transactions
- Fraudulent users, who try to impersonate legitimate users or make unauthorized transactions
- Attackers who try to hack into the website
- Attackers who try to disrupt the website operation through DDoS attacks



At a high-level Protector Air provides three core services:

a web security service
aesigned to filter all session activity for threats to the web application or website infrastructure

a session protection service,
designed to filter all session activity for threats to the web application or website infrastructure

a fraud tracking and alerting service
which monitors sessions and activity for indicators of Fraud

WEB SECURITY

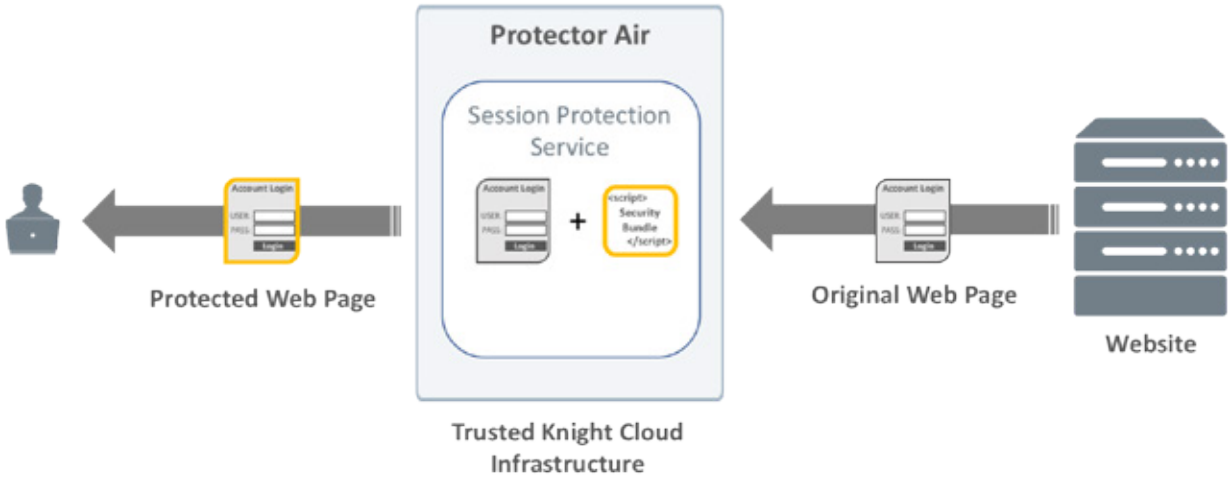
In addition to the other features, Trusted Knight has built full web application firewall (WAF) capabilities into Protector Air's Web Security Service. It provides a layer of defense in front of the web server, which can shut down probes, scan packets, and other undesired traffic — whether malicious or just irrelevant Internet noise. Protector Air will inspect all requests and block attacks on the web server infrastructure, software, and application.

With Protector Air, organizations can meet compliance requirements for web security, such as PCI DSS.
This not only includes any WAF requirements, such as mitigating OWASP Top Ten vulnerabilities, but also ensuring that the website uses best practices for communication, such as forcing all traffic to use HTTPS to prevent eavesdropping, avoiding older versions of SSL which are susceptible to man-in-the-middle attacks, etc.

USER AND SESSION PROTECTION

Protector Air layers protection on all web sessions to thwart malware that may be running on the user's computer or mobile device or within the application layer. It does this by a combination of page inspection and client-side security code that is added to web pages as they are delivered to users.

Because Protector Air is an in-stream solution, all web page content served by the website first passes through Protector Air before being forwarded on to the user's web browser. During this process, the Session Protection Service injects a bundle of javascript security code onto the web page. (This only affects the main HTML document – images and other content that is pulled in is unaltered.)



This protection is invisible to users – it does not affect the web page cosmetically at all, nor does it affect the functionality of the page.

This allows Protector Air to extend its security to the end-user side of the web session using several techniques, including:

- Identifying and disabling any javascript malware that attempts to run within the page
- Using techniques such as form encryption to foil attempts by malware running on the user's device to exfiltrate data, piggyback on sessions, hijack transactions, etc.
- Performing integrity calculations to identify if user-side malware is manipulating the web page content displayed to the user

FRAUD TRACKING AND ALERTING

Protector Air tracks indications of fraud by monitoring the interactions of the end-users. This includes:

- Indications that there is malware active on the user's device, browser, or web page that is attempting to interfere with the web session.
- Characteristics of the web session that indicate that the web session itself may be fraudulent, i.e. not initiated by a legitimate user

This information is tracked by Protector Air and allows an organization to know which users have likely been impacted by malware so that action can be taken (locking the user's account for example).

Note that Protector Air does not store any sensitive transaction data, such as account numbers, transaction details, credit card numbers, etc. For organizations that are additionally using larger fraud-monitoring solutions that do track and correlate this data, the intelligence from Protector Air can be fed into these systems to provide data for fraud analysis.

Protector Air Data Encryption

Data encryption is one of several techniques that Protector Air uses to defend against endpoint malware. Because malware such as banking trojans and keyloggers is running on the user's device itself it is not possible to disable the malware completely, like Protector Air can with javascript malware. Instead, Protector Air acts to prevent the malware from functioning as designed for fraud.

This malware will often act as a man-in-the-browser (MitB) to monitor all websites the user visits, stealing data such as usernames and passwords, payment card data, and other sensitive information entered by the user on websites. Most modern keyloggers work by intercepting form data as it is submitted - essentially grabbing a copy of the POST data the browser sends to the website.

Protector Air uses encryption to protect the form data within the form, so that the POST data the browser sends - and which the malware intercepts - is encrypted. This not only prevents the attacker from accessing the data, but also thwarts transaction hijacking, since any attempt by the malware to modify the data (e.g. by changing the recipient and amount of a payment) will fail on encrypted data.

Meanwhile, as the encrypted POST request passes back through Protector Air on its way to the website, it is decrypted and repackaged into its original form. Thus the website receives the form data in the format that is expected. Finally, because of this coordination between the user-side and the server-side protection, the attacker is also prevented from re-using encrypted data such as user credentials in a replay attack.

Conclusion

Organizations managing web applications need to consider the full scope of threats to the business. Full Transaction Stack Protection means extending protection out the end-user's device, ensuring the communication channel is resilient against service disruption, and defending the web server from targeted attacks and bots. It also encompasses all layers of the web application including the infrastructure and application layers as well as the transaction layer to defend against fraud.

Trusted Knight's Protector Air is the only unified solution for addressing security and fraud through Full Transaction Stack Protection. Furthermore, Protector Air's cloud-based, turnkey deployment means there is nothing to install or manage, and requires no integration or modification with the website. Finally, Protector Air has no software to download and zero impact on the user experience, eliminating user frustration and support headaches while still providing protection for 100% of the user base.

For more information

PLEASE CONTACT info@trustedknight.com | trustedknight.com/protector-air/

CONTENTS