10 Ways to Secure Your Digital Content

Today, digital security is top-of-mind. From the boardroom to the backroom, everyone is asking the same questions, “How do we protect our digital experiences? How do we ensure our website is safe for our visitors? How do we make sure that no one can steal our content?” In this whitepaper, we’ll discuss ten different ways to protect your digital content, ensure high availability, and maintain superior quality of experience for every digital visitor.

WHITE PAPER

Limelight
NETWORKS

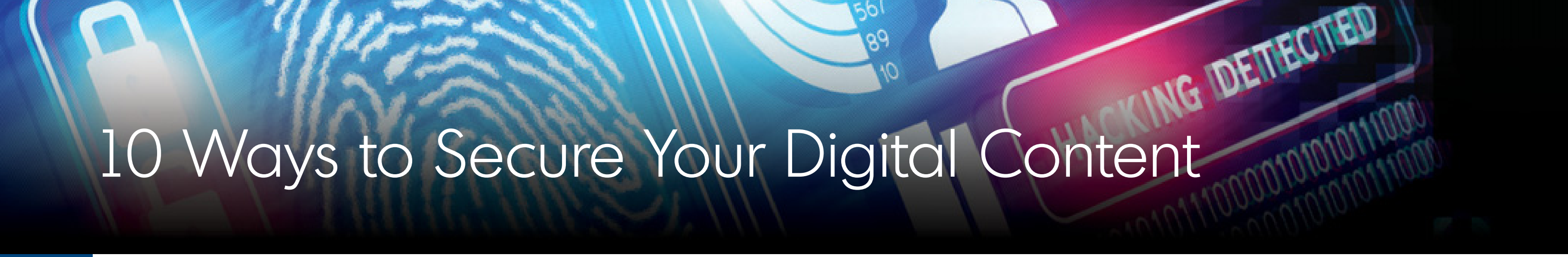
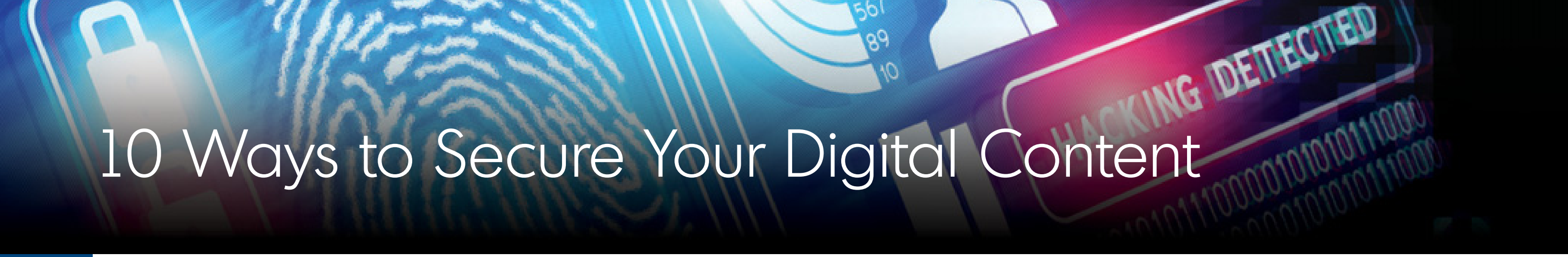


Table of Contents



Introduction

People are spending more time online everyday. In fact, according to Limelight's *2015 State of the User Experience* report, 45% of people are spending more than 15 hours a week online outside of work hours, up 95% from 2014.[1] This makes it ever more important to an organization that its digital experience is always available.

But that increasing time in the digital world isn't without recourse. Just as digital is becoming more important, it's also becoming a bigger target for attacks. According to a recent report[2], between 2013 and 2014, DDoS attacks rose 90% while the volume of those attacks grew as well—Radware's 2014-2015 *Global Application and Network Security Report*[3] indicated that over 20% of attacks are greater than 1GBps. Only it's not just DDoS attacks that are growing. According to Limelight Networks' *State of Digital Downloads*[4] consumer research report, people are increasingly okay with stealing digital content as well[5].

Whether it's a lack of availability because of an attack or theft of content, digital experiences need to be protected especially as they become the focal point for consumer engagement, interaction, and commerce.

Protecting Your Digital Content

Safeguarding a digital experience isn't a one-size-fits-all solution. It often involves multiple techniques and layers of security. We have identified ten different methods and technologies that an organization can employ to protect its content:

1. **Identity and Authentication**
    a. HTTPS
    b. User Authentication
2. **Content/Delivery Protection**
    a. Digital Rights Management (DRM)
    b. Tokenization
    c. Encryption
    d. Obfuscation
    e. Watermarking
3. **Access and Availability**
    a. Regionalized Access
    b. Distributed Denial of Service (DDoS) Protection
    c. Web Application Firewall (WAF)

Did You Know?

Limelight Networks has one of the largest HTTPS footprints on the planet. Early in 2014, Limelight merged its “general” HTTP pool with its HTTPS pool enabling them to not only serve HTTPS traffic from any POP in the network but to serve HTTPS at the same scale as it could normal, un-encrypted traffic.

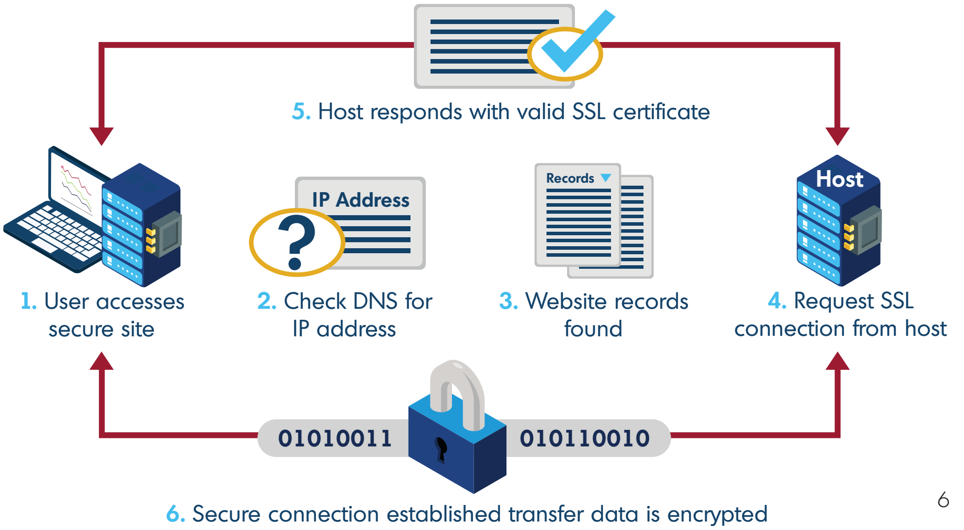
Additional Resources

- **HTTPS**—http://en.wikipedia.org/wiki/HTTPS

HTTPS

The very first, most fundamental level of protection you can offer is to assure your users they are accessing *your* digital experience. One way to accomplish that is to deliver your digital content over HTTPS.

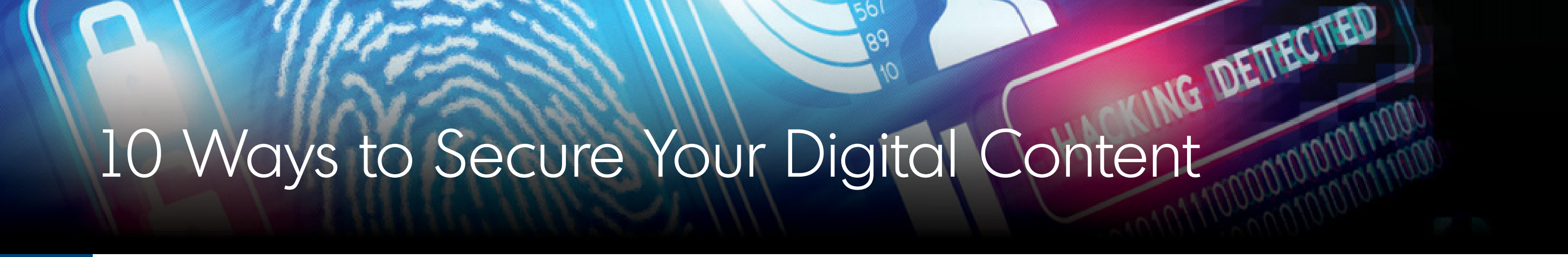
HTTPS, or HTTP-Secured, refers to the encryption of communication between a single client (typically a device with a web-browser) and the destination, such as a website, through a trusted certificate that verifies ownership of the destination. When a successful HTTPS connection is established, any data that passes over that connection (i.e., authentication credentials, a live video stream, etc.) is encrypted, protecting it from anyone who might intercept the transmission.


5. Host responds with valid SSL certificate
IP Address
Records
Host
1. User accesses secure site
2. Check DNS for IP address
3. Website records found
4. Request SSL connection from host
01010011
010110010
6. Secure connection established transfer data is encrypted

[6]

How does HTTPS work? The illustration above provides a simple explanation for an HTTPS connection initiated through a web browser. First, a user accesses the website or application and requests an HTTPS connection (1-4). Then, the website responds with a valid SSL certificate (5) that is tied specifically to the domain that the user is visiting. The user’s browser verifies the information in the certificate by comparing it with certificate information contained at a trusted third-party certificate holder and the domain name that they typed into the address bar. If everything checks out, a green lock is displayed in the user’s browser address bar indicating a secure, HTTPS connection (6).

HTTPS was primarily used by e-commerce companies to ensure a safe online shopping experience[7] but it is becoming increasingly adopted by organizations to ensure identity and to protect content—when the certificates between client and server are verified, the user can be certain with whom they are exchanging information.



Two-factor authentication

Two-factor authentication requires users to enter a randomly-generated, single-use number sent to them via email, text message, or on a special fob each time that they log-in. This ensures that even in the event a password is guessed or discovered, the perpetrator would still need to enter the code in order to gain access.

Additional Resources

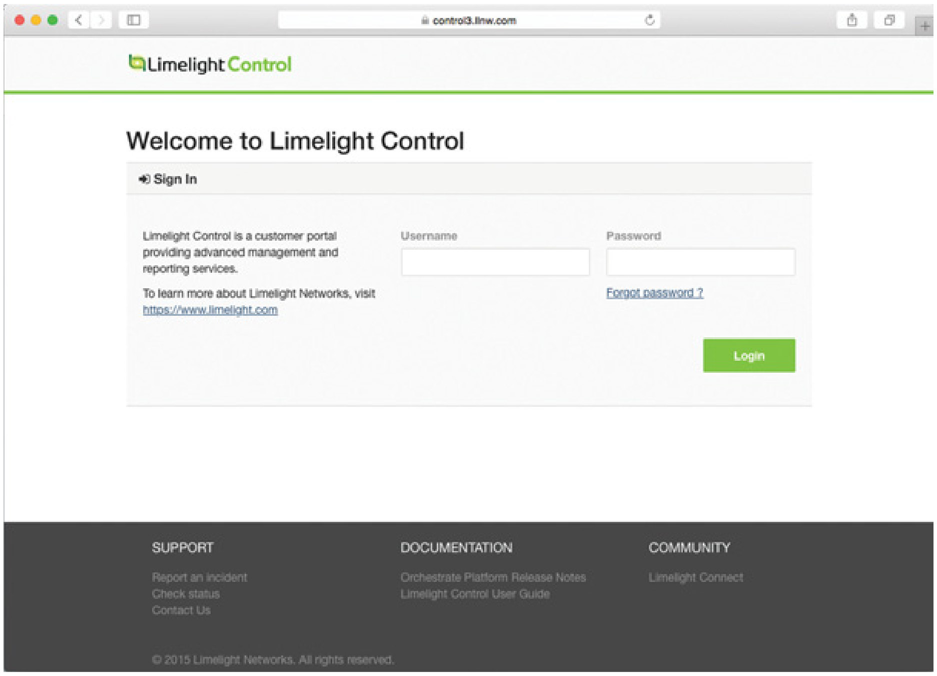
- **Two-factor Authentication**—https://en.wikipedia.org/wiki/Two-factor_authentication
- **CAPTCHA**—https://en.wikipedia.org/wiki/CAPTCHA

User Authentication

The simplest way to protect your digital experiences is to require that your users provide valid credentials to access specific areas or types of content. Depending on your website or web application, the required credentials can include a number of static and dynamic content elements:

- Username
- Password
- Personal Identifiable Information (PII) like the last 4 digits of a social security number
- Answers to security questions
- CAPTCHA

Server-side scripting authenticates the user by processing these data elements. Valid users are provided access, invalid users are turned away.



Welcome to Limelight Control

Sign In

Limelight Control is a customer portal providing advanced management and reporting services.

To learn more about Limelight Networks, visit https://www.limelight.com

Username

Password

Forgot password ?

SUPPORT
Report an incident
Check status
Contact Us

DOCUMENTATION
Orchestrate Platform Release Notes
Limelight Control User Guide

COMMUNITY
Limelight Connect

© 2015 Limelight Networks. All rights reserved.

As illustrated above, user authentication is a simple way to keep content secure. Limelight Control also uses a two-factor authentication mechanism to ensure the tightest level of security for accessing sensitive data and account functionality.

DRM Packages

A DRM package refers to how the content is encrypted for playback. Packages often require a specific player in order to decrypt the content. Some of today's more popular DRM formats include:

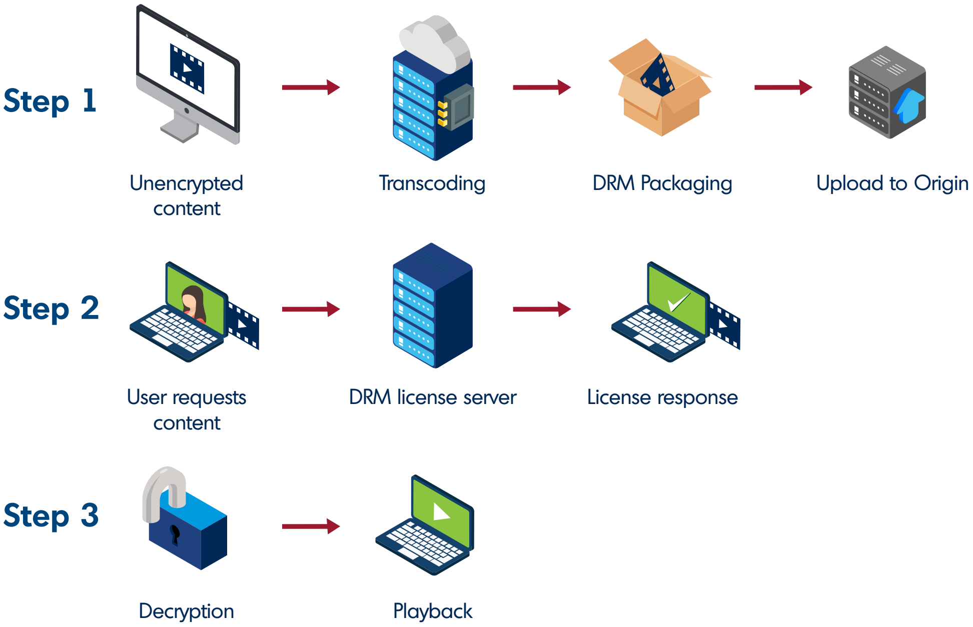
- Microsoft PlayReady
- Widevine
- Adobe Access
- Marlin
- Ultraviolet[8]

Additional Resources

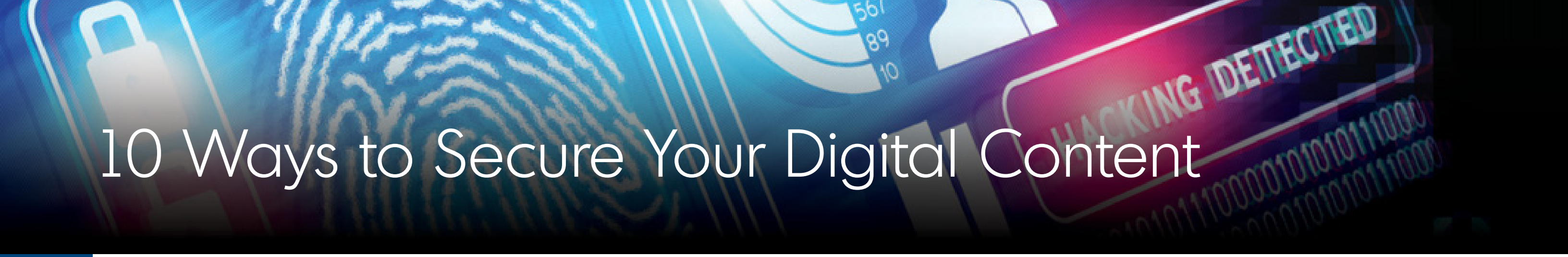
- **Digital Rights Management**—http://en.wikipedia.org/wiki/Digital_rights_management
- **How Stuff Works**—http://computer.howstuffworks.com/drm.htm
- **Microsoft PlayReady**—https://www.microsoft.com/playready/
- **Google Widevine**—http://www.widevine.com/
- **Adobe Access**—http://www.adobe.com/products/adobe-access.html
- **Marlin**—http://en.wikipedia.org/wiki/Marlin_%28DRM%29
- **Ultraviolet**—http://en.wikipedia.org/wiki/UltraViolet_%28system%29

Digital Rights Management (DRM)

When part of your digital experience is video, you may be required to protect that content by preventing unauthorized people from viewing. For example, if you are licensing the material from a third-party they may want to ensure that their videos can't be stolen and redistributed on the Web. And although technologies to prevent deep linking can help, they won't prevent against authorized users posting online or sharing the content they have purchased. That's where DRM comes into play.



In Step 1, unencrypted content is sent through a packager and uploaded to origin. In Step 2, a user requests the packaged content (by clicking a link, making a purchase, selecting in an application, etc.). The application or website the user employs to access the content requests a license from a DRM license server to decrypt the content by passing along key information about the user. The license server checks the incoming request (and user information) against its database and passes back a response to the player. Finally, in Step 3, if the information is valid and the user has viewing rights, the content is decrypted by the client for playback.



Kinds of encryption?

There are a host of common algorithms used for encryption which can be implemented from most programming languages:

- Triple DES
- RSA
- Blowfish
- Twofish
- AES
- MD5

Additional Resources

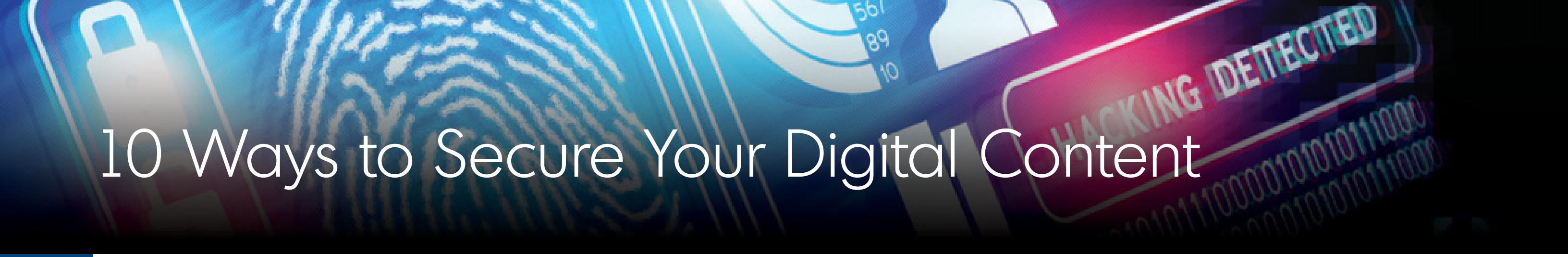
- **What is Encryption?** http://www.webopedia.com/TERM/E/encryption.html
- **Guide to Cryptography**—https://www.owasp.org/index.php/Guide_to_Cryptography
- **Cryptography**—http://en.wikipedia.org/wiki/Cryptography

Encryption

Over the past decade, websites have changed dramatically. Once just static text and images, today's websites are dynamic and highly personalized. They integrate with third-party services like Facebook and Twitter, include targeted advertisements, and can even be reshaped based on user history.

The result of all this is that websites have become more dependent upon back-end databases to enable this high-end functionality. And therein lays the problem. Some of that stored data can be "personally-identifiable information" (PII) such as names, addresses, emails, and credit card numbers. It needs to be protected, not just stored in some row in the database. It has to be encrypted.

Thankfully, there are lots of programmatic ways that you can encrypt your data. Using server-side scripting, for example, you can code part of your web pages to do the encrypting and decrypting on the fly, ensuring that data put into the database, especially PII, is secured even if a breach does occur.



Tips and Tricks for Obfuscation

There are a couple of other recommendations that you should consider when employing this method of content protection:

- **Store content above the WWW**—you should store sensitive content (like media files) above your www directory. This prevents anyone from crawling the website to find the location of your content. For example, let's say you have twenty directories off the root / of your web server one of which is /www/ and holds all of your website content. Instead of /www/dirA/dirB/dirC/content.file, the content would be stored in one of the other directories off the root / and accessed via explicit URI.
- **URLs in the database**—you should store URLs to your content in a database rather than hard coding them directly into a page with a link. That way, if someone views the code on your page, they can't find the content location.
- **Prevent directory listing**—on Apache webservers you can prevent people from listing directory contents by adding a .html file to each directory or turning off indexing in the .htaccess file[9].

Obfuscation

There's not much worse than spending a lot of time, effort, and money to create content for your digital experiences only to have that content displayed on another website without your permission. But without a way to protect the location of your content, that's exactly what can happen. It's called "content scraping" and "deep linking."

There are a host of methods you can use to conceal the location of your content but the best way to do that is to mask its location using a server-side script. Consider the two URLs below*:

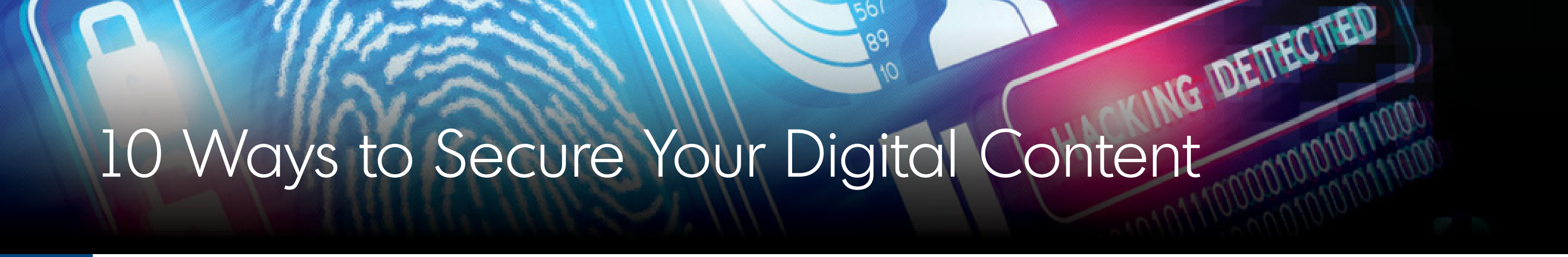
http://content.mycompany.com/dirA/dirB/dirC/contnetname.mp4

http://content.mycompany.com/dirA/dirB/dirC/contentname.php

*The URL is for illustrative purposes only.

In the first example, the content location is clearly identifiable as a .mp4 file (which can be scraped from a page using a variety of software). But in the second, the content location is obfuscated by referring the request to a server-side script (contentname.php) that processes and returns the result directly to the browser without a visible URL.

The result, although simple in its implementation, can help protect the location of your content from prying eyes.



Tokenization

A more advanced form of obfuscation, tokenization, refers to "the process of substituting a sensitive data element with a non-sensitive equivalent, referred to as a token, that has no extrinsic or exploitable meaning or value. The token is a reference (i.e. identifier) that maps back to the sensitive data through a tokenization system. The mapping from original data to a token uses methods which render tokens infeasible to reverse in the absence of the tokenization system."[10]

In a tokenization system, parameters on the URL are used to determine the authenticity of a request. If the request is not "authentic" (that is, it doesn't have a valid token), it is denied, most often resulting in a HTTP 401 error.

Consider the following*:

http://www.website.com/media/video.mp4?e=1427821200&ip=81.0.10.0/24&r=www.website.com&h=7ff066fa7b8710e6e26b2bcb1435d969

* The URL is for illustrative purposes only.

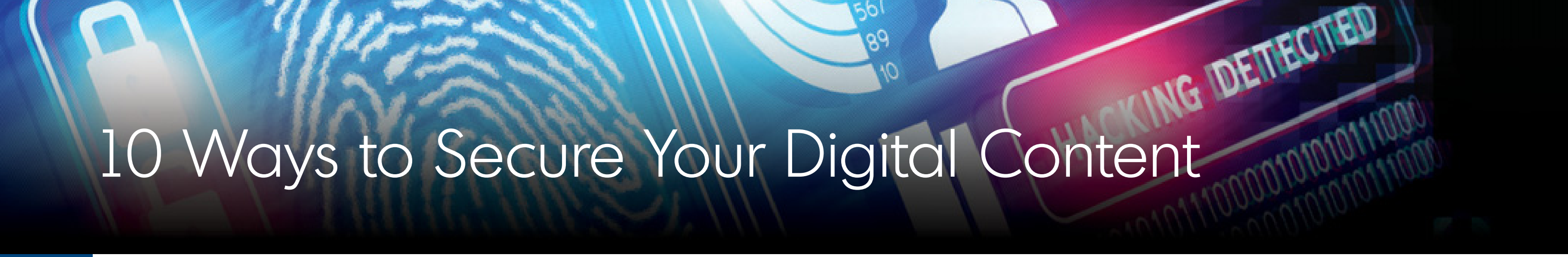
How does it work? Each parameter passed on the URL is employed as part of the tokenization system:

- e—this represents the time the token expires
- ip—accepted IP range or individual IP
- r—the referring URL
- h—the token

The token, "h," is created using the referring URL "r," the time the token expires "e," the requesting ip address, and a shared secret. When the requesting URL is received, the tokenization system attempts to recreate the token using the parameters. If the token can be re-created, the request is considered authentic and valid and the content is served to the requesting user.

Additional Resources

- **Tokenization**—http://en.wikipedia.org/wiki/Tokenization_%28data_security%29
- **Apache tokenizer**—https://cwiki.apache.org/confluence/display/solr/Tokenizers

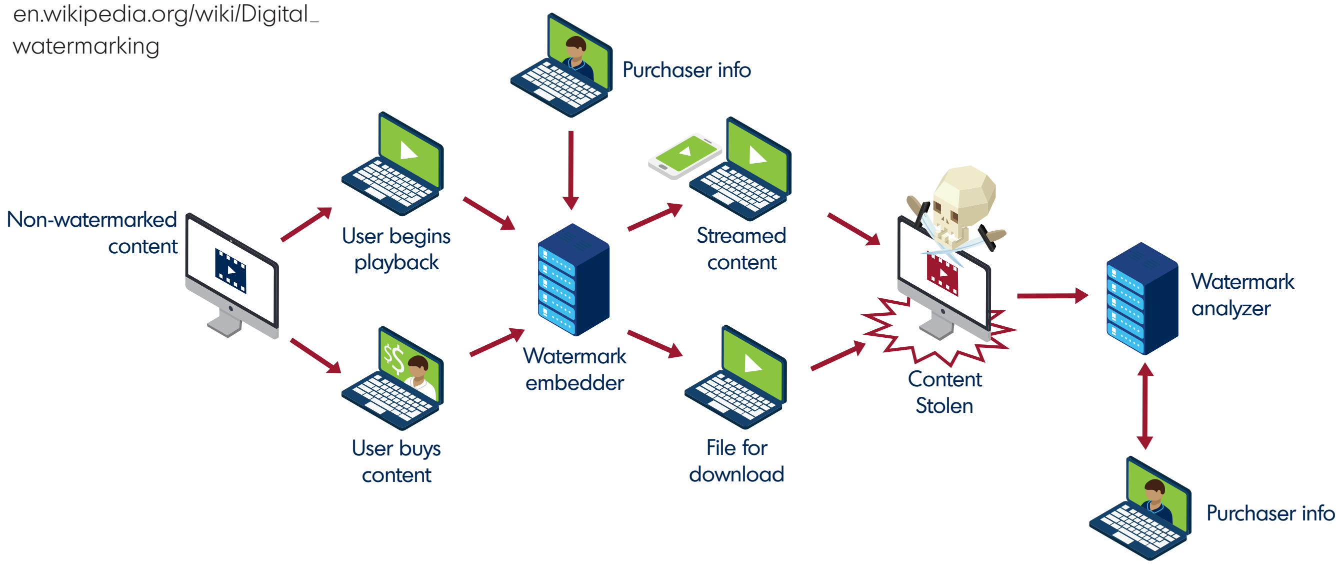


Additional Resources

- **Verimatrix**—http://www.verimatrix.com/solutions/forensic-watermarking
- **NexGuard**—http://www.nexguard.com/forensic-watermarking-introduction/
- **Definition**—http://whatis.techtarget.com/definition/forensic-watermark-digital-watermark
- **Watermarking**—http://en.wikipedia.org/wiki/Digital_watermarking

Watermarking

How do you prevent people from sharing your videos on the Internet? One way is to tag them with an invisible watermark. In many forms of watermarking, the actual mark is dynamically generated and can include identifying information like IP address, purchaser's name, and even the last four digits of a credit card number. If a copy of the video is found on the Internet, special software can "decode" the watermark to identify the original purchaser. Many hotels use this technology to prevent people from recording a pay-per-view movie they watch in their room and then sharing it—the watermark is dynamically generated to include information about the room number and time of day the movie was watched!



How does watermarking work? The content is run through a watermark embedder that takes metadata about the purchaser such as location, IP address, purchase day and time, etc., and inserts it into the video as a binary string that represents the data. The 1s and 0s of the string are implemented in sequencing frames by tagging one or more pixels. If a pirated copy of the content is found on the Internet, for example, software can retrieve the binary string by analyzing the frames and decoding the pixels. Once reassembled, the binary string can be converted back into the metadata so that the identifying information can be retrieved.

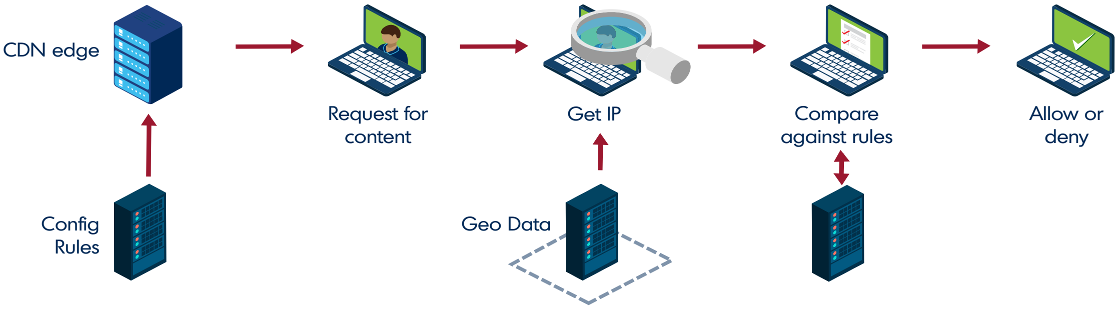
Additional Resources

- **Neustar IP Intelligence and Geolocation services**—https://www.neustar.biz/services/ip-intelligence
- **IP2Location GeoLocation Database**—http://lite.ip2location.com/

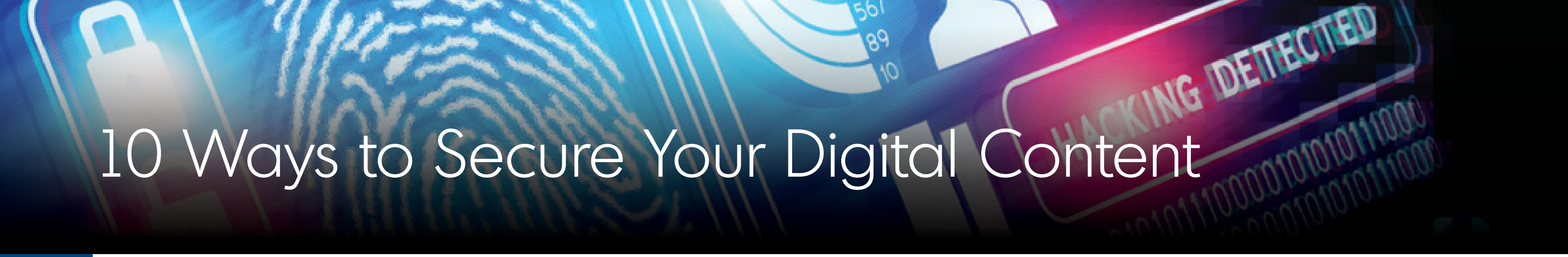
Regional Access

In some cases, such as when licensing content from a third party (like video), you may be required to restrict access to specific geographic regions: visitors from outside of the regions won't be able to view the content. In the digital content delivery world this security mechanism is known as geo-fencing and refers to the use of geographical data (i.e., world region, country, zipcode, etc.) to either allow or deny access to specific content.

When a request for the content is received, the IP address is used to reveal the geographic location of the user. If the location falls into a whitelist or blacklist of geographies, the request can be granted or denied respectively.



The way geo-fencing works is relatively simple. When a user request is made to the CDN edge, the IP address of the user's location is taken from the request header. A business-rules engine then compares that IP against a database of geo-locations. Based on that, the user's request for content is either approved or denied. If it's denied, the user can be presented with a webpage or message that indicates the content is not available in their area.



Why Security in the Cloud?

Cloud-based DDoS security actually has a number of advantages over on-premise equipment:

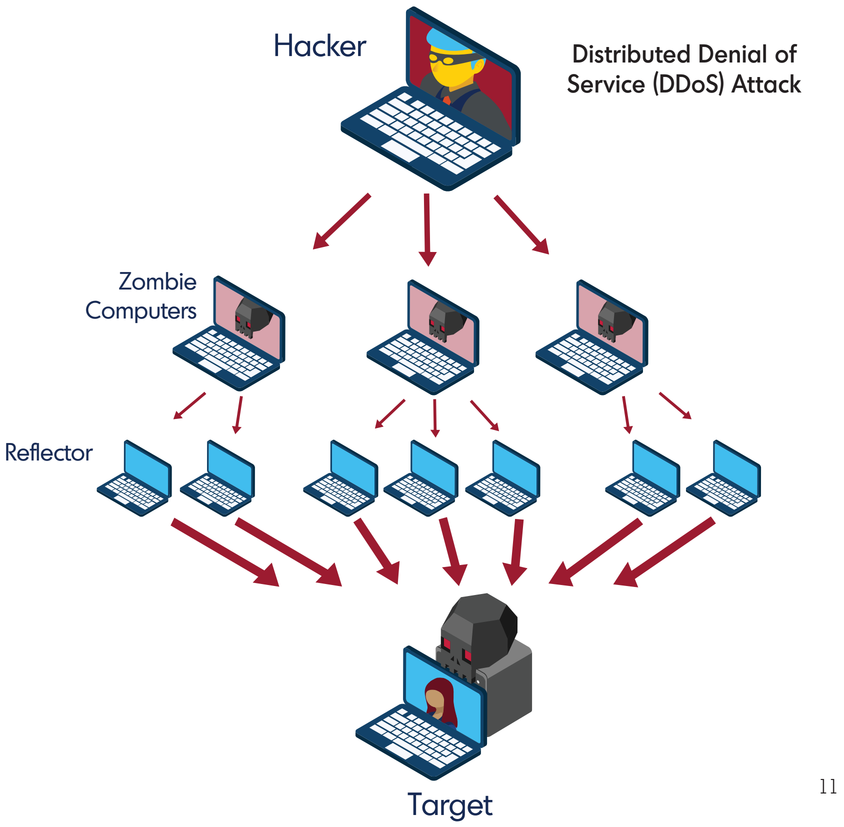
- **Upstream**—if you are already using a CDN provider to deliver your digital content, detecting and mitigating an attack can come at the network edge, potentially thousands of miles from origin thereby sparing your network from any potential fallout or impact. When combined with scrubbing, only good traffic is returned to origin preventing your bandwidth from being flooded with bad traffic.
- **Absorption**—as a distributed network, most CDNs have thousands of servers against which they can spread out an attack, even preventing Layer 3 and Layer 4 attacks (two common DDoS vectors) from ever reaching the origin.
- **Resiliency**—with those thousands of servers and terabits of egress capacity, the CDN quickly returns to normal operations in the wake of volumetric DDoS attacks. Even while under duress, the CDN can still continue to provide accelerated content delivery.

Additional Resources

- **What is DDoS?**—https://en.wikipedia.org/wiki/Denial-of-service_attack
- **Digital Attack Map**—http://www.digitalattackmap.com/understanding-ddos/

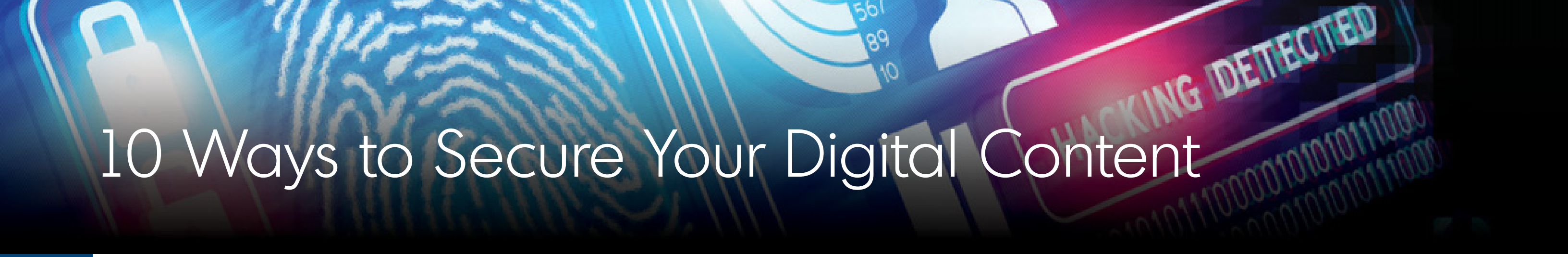
DDoS Protection

Ensuring your digital content and experiences are available is just as important as encrypting sensitive data or protecting against theft. And with the rising number of attacks, it's critical that you enable a layer of security in front of your origin that mitigates the potential of Distribued Denial of Service (DDoS) attacks.



What happens when an attack occurs? It depends upon the layer, and nature, of the attack. If the attack is in layer 4 of the networking stack, a botnet sends a flood of ping or requests on port 80 to the target webserver. The increase in traffic takes up valuable bandwidth so that the webserver can no longer effectively serve legitimate user requests. If the attack is in layer 7, a botnet may attack a specific webpage by impersonating user behavior (i.e., a login form) and taking up valuable computer resources by continually submitting bogus information until the webserver is no longer able to serve legitimate user requests.

There are a number of different options available for this layer of security. You can host on-premise equipment in front of your origin. You can rely on your ISP or your datacenter (if you are hosting your site elsewhere). Or you can employ cloud-based services.



What's a WAF?

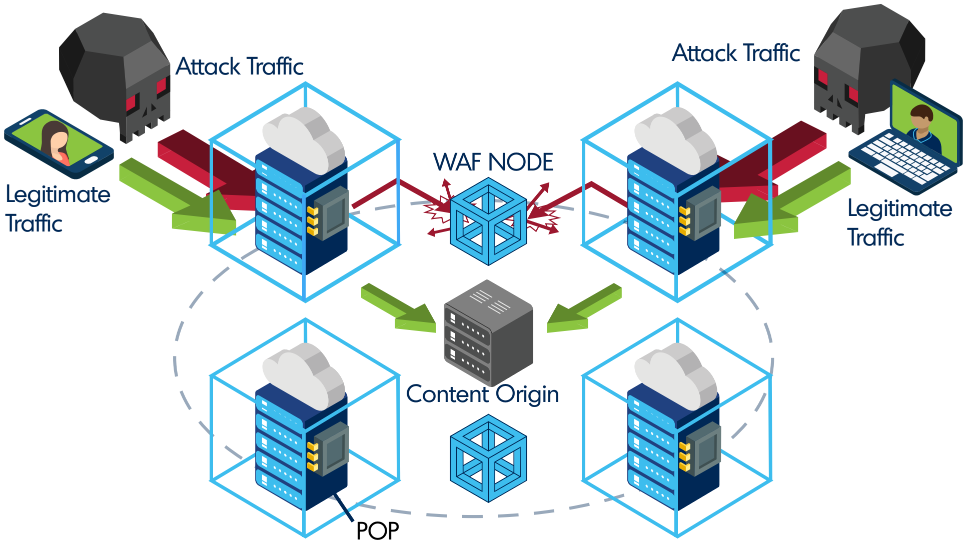
A web application firewall (WAF) is an appliance, server plugin, or filter that applies a set of rules to an HTTP conversation. Generally, these rules cover common attacks such as cross-site scripting (XSS) and SQL injection. By customizing the rules to your application, many attacks can be identified and blocked. The effort to perform this customization can be significant and needs to be maintained as the application is modified[12].

Additional Resources

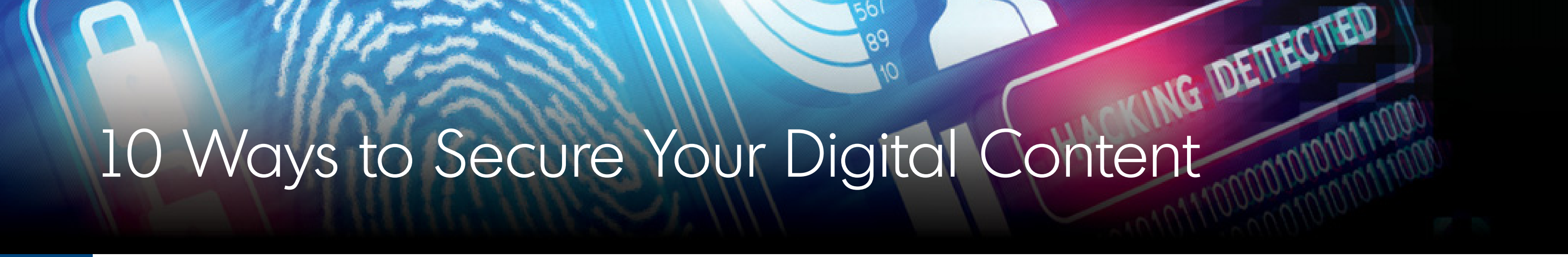
- **What is a WAF?**—https://en.wikipedia.org/wiki/Application_firewall

Web Application Firewall

The sophistication of cyber-attacks sometimes warrants a layering of security technologies in front of a website. For example, where DDoS protection can help prevent a flood of malicious traffic, a WAF, or web application firewall, can help filter traffic against a set of rules to prevent more targeted activity like cross-site scripting (XSS) and SQL injections.



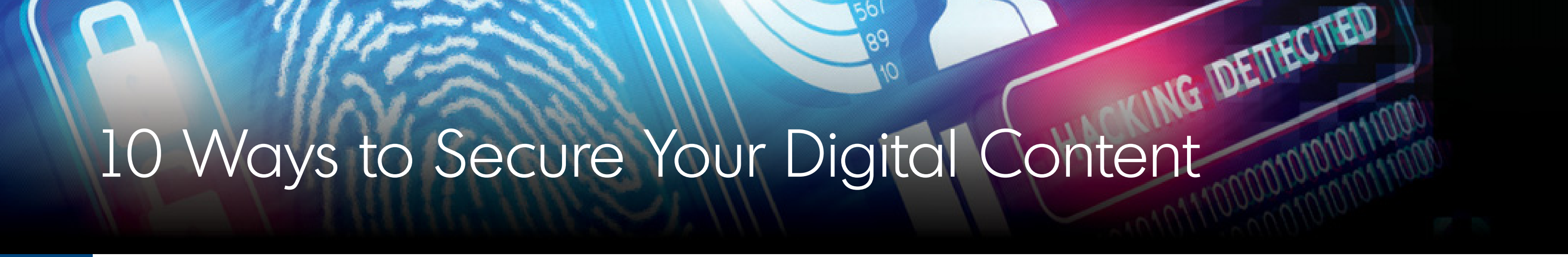
So how does it work? Web application security is best achieved with globally distributed infrastructure. This implementation leverages the CDN as an integral part of protecting websites. WAF nodes are located between origin servers and the CDN. The CDN does all the heavy lifting – caching, acceleration, and delivery of static content to websites. The CDN forwards only dynamic content to the WAF nodes, as this is the form attacks take, so the origin websites are shielded by the CDN and traffic filtering by the WAFs. After WAF evaluation, the traffic is either passed onto origin (which is configured to only accept requests from the WAF) or denied. The result is complete lockdown of IP traffic – the WAFs only accept traffic from CDN nodes, eliminating sacrificing performance for security.



Limelight Cloud Security Services

Cloud Security is a comprehensive set of content and infrastructure protection tools designed to ensure that your digital experiences are always available:

- **DDoS Attack Detection and Mitigation**—integrating a WAF solution with the Limelight Content Delivery Network (CDN) is the perfect combination to combat the sophistication of today's attacks, providing a protective shield around your security perimeter. Leveraging our cloud-based network resources delivers cost effective protection for your web applications without a performance penalty.
- **Web Application Firewall**—integrating a WAF solution with the Limelight Content Delivery Network (CDN) is the perfect combination to combat the sophistication of today's attacks, providing a protective shield around your security perimeter. Leveraging our cloud-based network resources delivers cost effective protection for your web applications without a performance penalty.
- **MediaVault**—integrated support for a tokenization system for content access authorization.
- **IP Whitelisting and Blacklisting**—supports IP lists to control content access.
- **CORS Management**—Add, Set, and override Cross Origin Resource Sharing (CORS) directives on a per-request basis, allowing multiple content origins while restricting resource sharing to designated origins.
- **DRM**—current support of Widevine and Microsoft PlayReady provides protection against content theft.
- **Content Encryption (SSL)**—Limelight Networks has a global SSL footprint that can deliver encrypted HTTP traffic at scale anywhere in the world, to any device.
- **Customer Certificates**—support for hosting existing customer SSL certificates.
- **Geo-Fencing**—integrated within the Limelight CDN software, geo-fencing enables you to lockdown content accessibility by region, by country, or by zip/postal code.
- **RTMPE**—the secure Real-Time Messaging Protocol Encrypted supports Flash traffic for high-performance transmission of audio, video, and data streams.



Conclusion

Securing your digital experiences is critical to ensuring the best possible user experience. From verifying your identity (with HTTPS) to encrypting sensitive data to restricting access and protecting multimedia content, you must approach security in a layered manner, employing multiple means and techniques to protect the digital content through which your audience interacts.

Limelight Orchestrate Security provides a host of technologies—such as HTTPS, DDoS mitigation, geographic restriction, and DRM—that you can employ alongside tried-and-true techniques like data encryption and authentication to build a highly-customized defensive strategy against the increasing attacks permeating the digital world.

[1] http://img03.en25.com/Web/LLNW/%7B63aeeb04-0509-41a8-b638-9d6cb6f8a5a4%7D_StateofUser_102015.pdf

[2] http://www.zdnet.com/article/global-ddos-attacks-increase-90-percent-on-last-year/

[3] http://www.radware.com/ert-report-2014/

[4] http://img03.en25.com/Web/LLNW/%7B28353ac8-9905-4865-9207-4ccfcf744330%7D_StateofDigital_WP_8.5x11_Web.pdf?ls=SCDPR

[5] According to the study, about 12 percent of respondents were okay with stealing movies and TV shows. That number climbed to 20% for Millennials.

[6] https://hilfans.wordpress.com/2014/09/30/manfaat-koneksi-ssl-untuk-sebuah-website/

[7] This was accomplished by encrypting credit card data in transit between a user's browser and the e-commerce website.

[8] Not a specific DRM package (as it supports multiple DRMs) but worth including

[9] http://en.wikipedia.org/wiki/Tokenization_%28data_security%29

[10] http://www.thesitewizard.com/apache/prevent-directory-listing-htaccess.shtml

[11] http://computer.howstuffworks.com/zombie-computer3.htm

[12] https://www.owasp.org/index.php/Web_Application_Firewall

Limelight NETWORKS

Global Headquarters
+1 602.850.5000

Europe, Middle East & Africa
+44 203 728 6300

Asia Pacific Region
+65 6829 7125

info@limelight.com ▪ limelight.com