Protect Your Project Management Applications from Common Attacks

LOADSPRING

Table of Contents

187 Ballardvale Street, Suite B-210, Wilmington, MA 01887 +1 978.685.9715 information@loadspring.com

Introduction

Data security is no longer a luxury, but a critical necessity. By now we have all seen or heard that not having the right security measures in place can impact a company with massive financial and reputation losses. For example, a leading security vendor reported over 63 million attack attempts over one week, of which 17 million were directed against the US and 10 million against Canada, which are two of the major regions that LoadSpring™ operates in. In the US alone, it is estimated that over $100 billion USD are spent every year in security costs, as well as exploits, and costs to recover from attacks.

The type of attacks companies face differs by industry. Although the most common attacks are SQL injection, cross site scripting, brute force, DDoS, SSL Vulnerabilities, and ransomware, the most prevalent attacks specifically in the field of Cloud Project Management are SQL injection and cross-site scripting.

SQL Injection

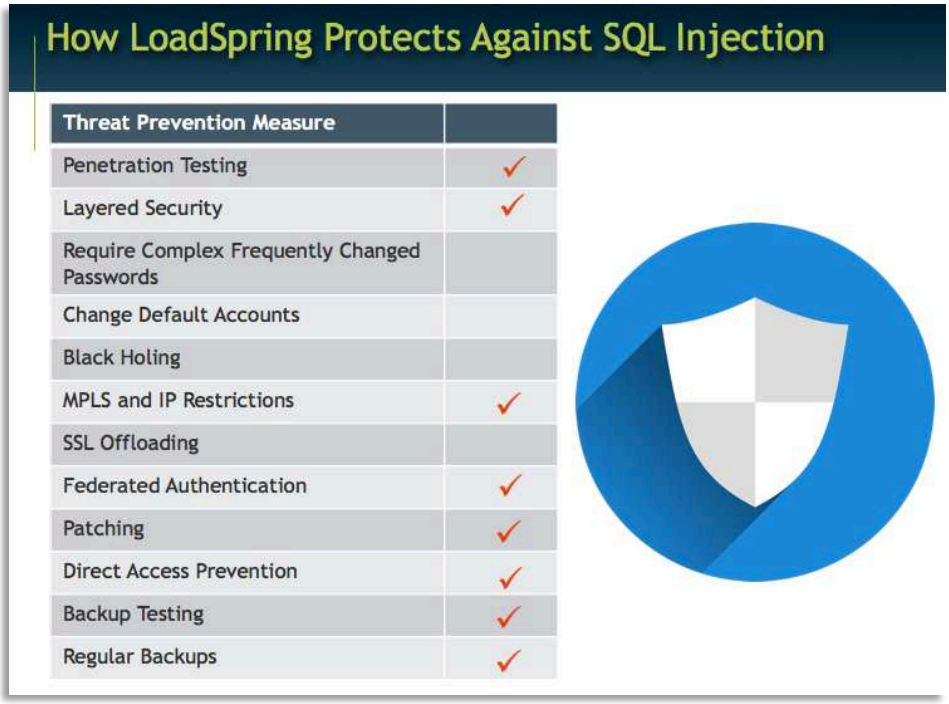
A SQL injection is a technique that is used to attack a database driven application by inserting SQL statements into an input field expecting to get input back. Once the attackers get the input, they execute specific statements that allow them to extract or inject data into the database. The statement may be as simple as, if 1=1 then do *this*, which would allow the attacker to either pull or inject table information, depending on the type of SQL vulnerability. This gives them the ability to steal, tamper, or expose the user's identity.

One example is the Barracuda Networks breach. Ironically, they are a security vendor that develops a number of products to test or protect against SQL injections. This attack exposed their users, along with their user passwords and confidential client information. In an attempt to build their credibility back Barracuda exposed their mishaps, one of them being that they momentarily turned off some of the application firewalls protecting them against this type of attack.

Another example is the 2016 attack on the Illinois election, where attackers exposed the personal information of over 200,000 voters.

SQL injection attacks pose a high risk to the confidentiality and integrity of your data.

How does LoadSpring protect against SQL injection attacks?



It is critical for companies to have an onion type approach to security in order to minimize risk, being that having only one measure in place will most likely not prevent an attack. Attacks are evolving constantly, so it is difficult to stay a step ahead of the attackers, which is why putting multiple layers in place to protect against attacks and common security mistakes is imperative.

- **Penetration testing** is one of our most common prevention measures ran both by ourselves and with third-party vendors. The latter help us test and validate the applications we write, as well as the third-party applications we host. These results disclose vulnerabilities and patches, in order for us to take immediate action.

- Some companies will only allow connectivity to their applications over **Multiprotocol Label Switching (MPLS),** which is a dedicated network that specifically goes from their offices to our office via strict general internet access to their applications.

- Identifying **patches** and making sure they're up to date is critical.

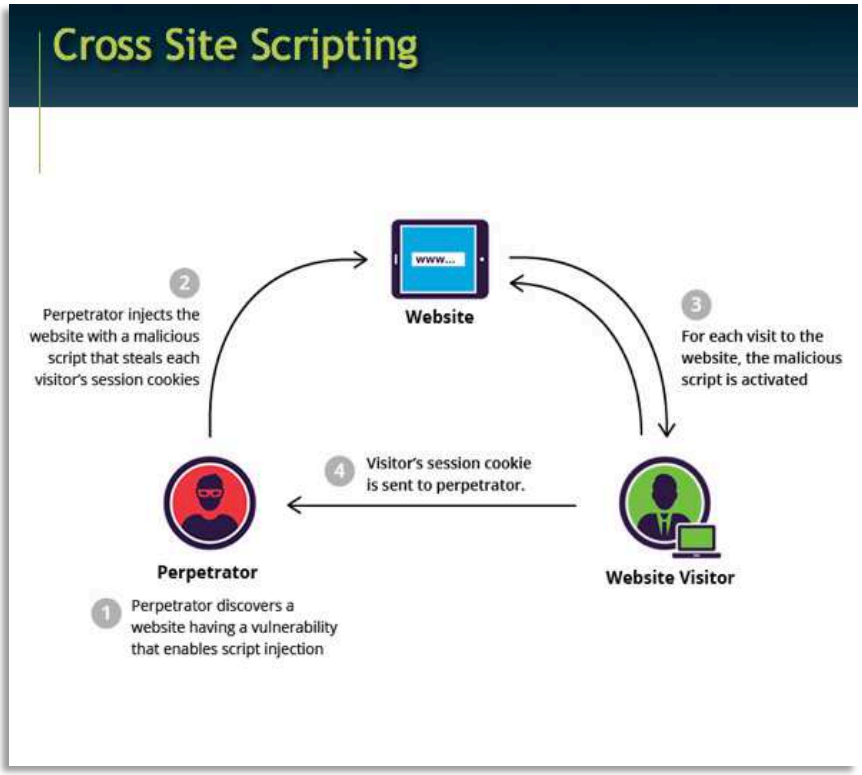
- **Direct access prevention** blocks attackers from indexing and accessing information that lacks complex user authentication. This means attackers must know the URL of the application they are attacking since they are not assigned directly to an IP. Direct access prevention is important because attackers run random scans against IP addresses and then identify vulnerabilities. So, if we have clients hosting older applications that may not be up to date, or for which the SaaS vendor has not provided a patch, it's critical that we have other layers of security in place. This issue stands out more often in the Construction and Engineering industries where a specific version of an application must to be used through the full life of a project, regardless of upgrades available.

Cross-site Scripting (XSS)

Cross-site scripting (XSS) is a client-side code attack where the attacker injects malicious code into a website. This type of attack is more common with applications that require user input.

An XSS attack happened recently with Equifax, where users who visited the Equifax website would unknowingly download malware.

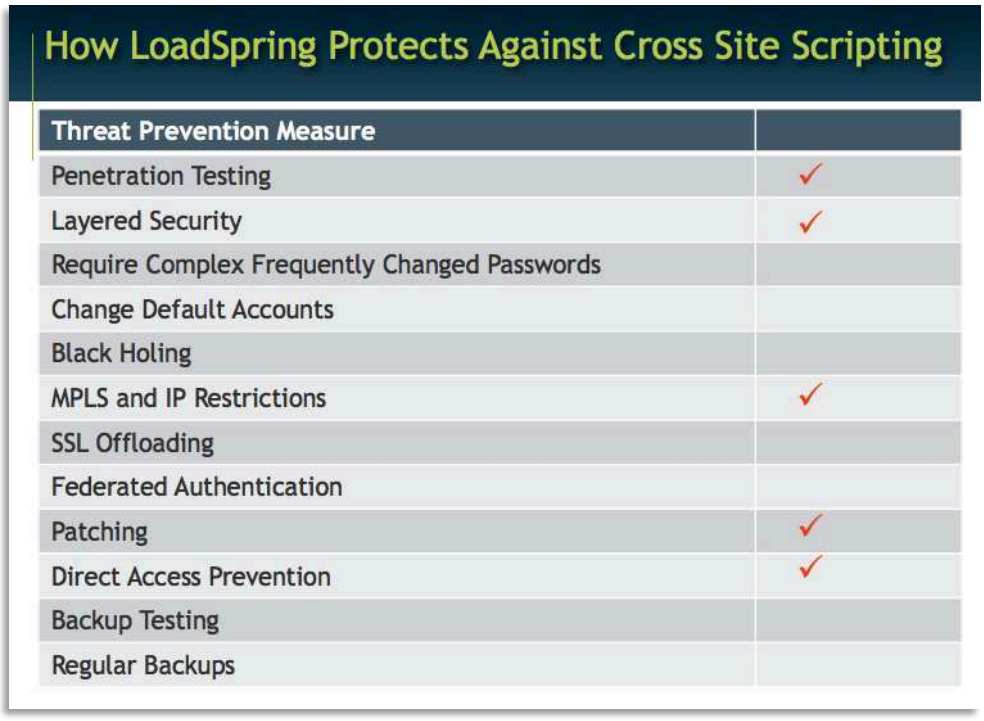
One of the biggest risks of an XSS attack is that it allows the attacker to take the session cookie from the end user. This allows the attacker to spoof that user session while granting them user access either into the application or that user's credentials, which would then allow the attacker to authenticate and act as a normal user and gain full access.



The biggest risks in XSS attacks are the loss in confidentiality and integrity of your data being that attackers would gain access and can enter bogus data.

How does LoadSpring protect against XSS attacks?

Again, it is going to be through a mixture of different security layers.



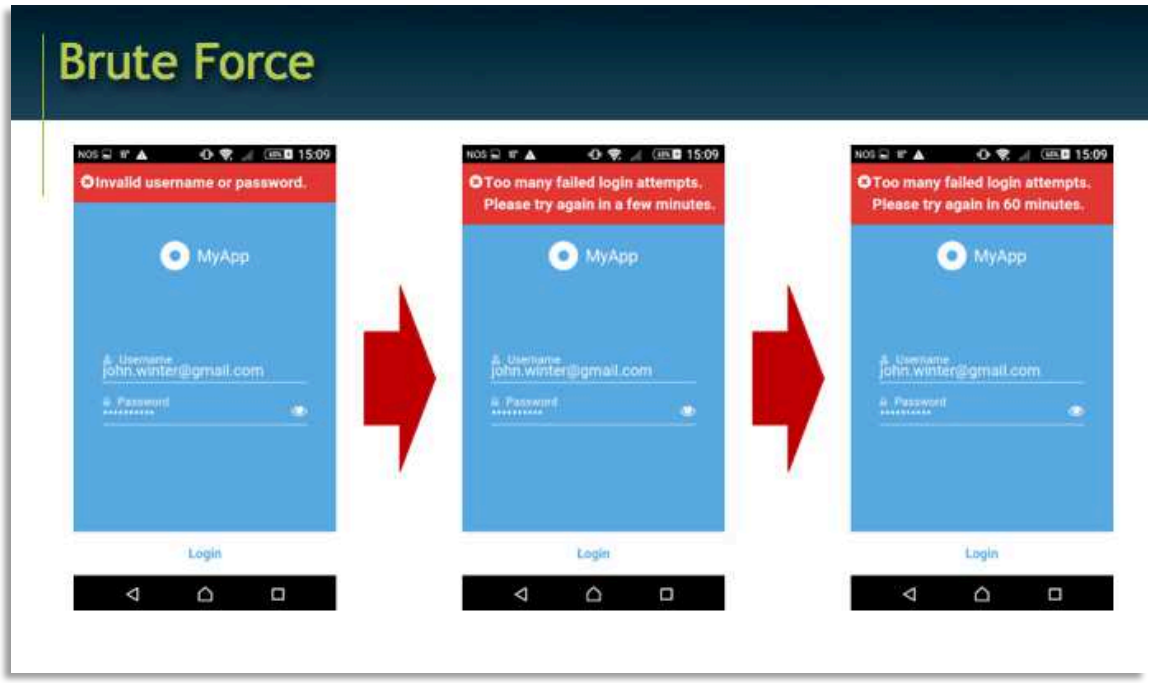
How LoadSpring Protects Against Cross Site Scripting

Threat Prevention Measure	
Penetration Testing	✓
Layered Security	✓
Require Complex Frequently Changed Passwords	
Change Default Accounts	
Black Holing	
MPLS and IP Restrictions	✓
SSL Offloading	
Federated Authentication	
Patching	✓
Direct Access Prevention	✓
Backup Testing	
Regular Backups	

- **Penetration testing.** We often run simulated XSS attacks against our websites to identify whether they are vulnerable or not. If they are, then we work with our vendors and developers to protect against that threat.
- We place **layered security**, so that attackers can't get to sensitive information.
- We use **Internet Protocol (IP) restrictions** to block attackers from accessing the application.
- **Patching**
- **Direct access prevention**

Aside from these five main measures to protect against an XSS attack, we randomly execute an Intrusion Prevention System and an Intrusion Detection System.

Brute Force



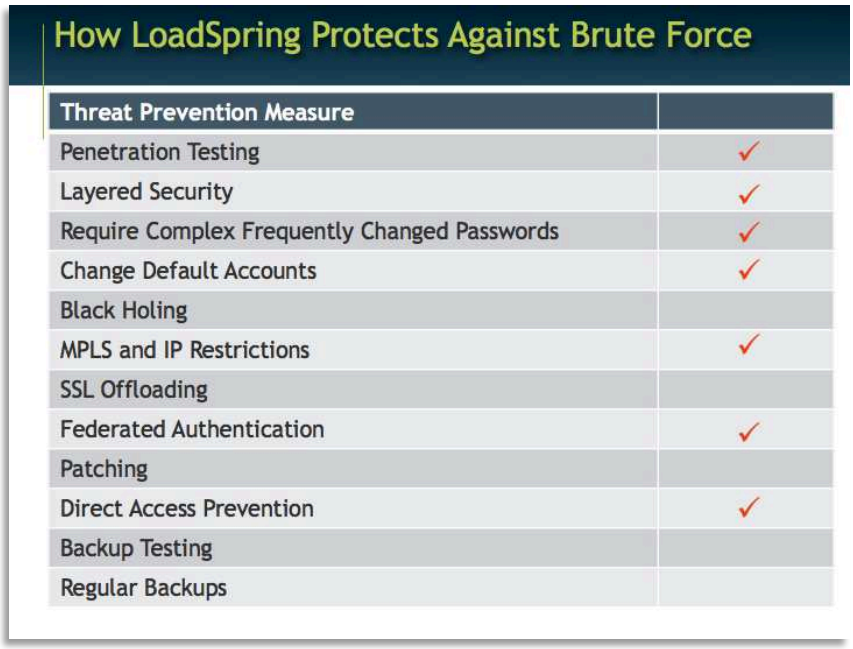
Generally, a brute force attack is not a software vulnerability, but it targets weak passwords and common accounts. Companies can detect this type of attack through an unusual spike in memory or CPU usage of an application, meaning if an attacker is hammering an application it will impact its overall usability. The risk here is that, if an account is compromised, the attacker can impersonate the user and obtain full access to data.

Shire Corporation experienced a brute force attack when one of their Construction Managers logged into the Rhode Island Department of Transportation's database using a weak username and password to get information from other vendors that were bidding against them at projects. Doing so gave Shire Corp. a very strong competitive advantage since they were able to underbid and provide information directly, unlike other bidders that didn't have insider information.

Another example of a brute force attack was the Target incident back in 2013 where the attackers gained access to vendor information, which compromised an application that had elevated privileges in Target's network. This allowed the attackers to install malware on all payment terminals, which executed a man-in-the-middle attack ultimately obtaining shopper credit card information.

How does LoadSpring protect against brute force attacks?

As with all types of attacks, it's important to rely on several preventative measures to protect against a man-in-the-middle or brute force attack.



- **Penetration testing** helps detect and protect against vulnerabilities, such as locking accounts after a certain number of failed attempts to access. This slows down the ability of an attacker to execute a brute force attack.

- **Layered security**

- **Requiring complex and frequently changed passwords,** versus simple passwords such as "password" or "123456" enables us to protect our customers.

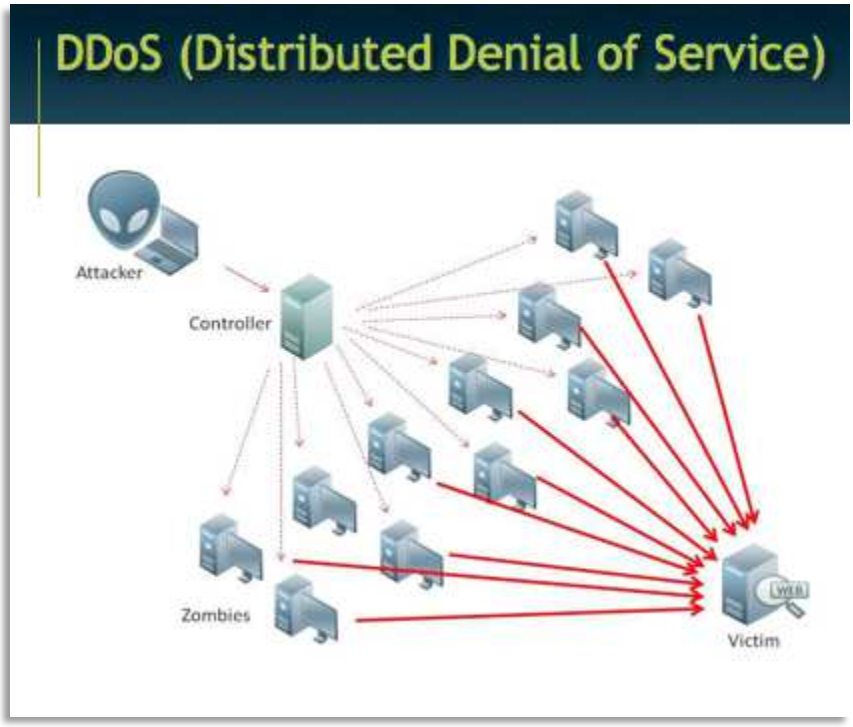
- **Changing the default accounts**. Some applications like Primavera P6 have common accounts that have very simple default passwords, which should be updated. If possible, make sure those accounts are disabled in order to create your own accounts, or change the username to something more complex.

- **MPLS and IP restrictions**

- **Federated Authentication** is another measure we take to make sure applications remain secure. With federated authentication, we connect to the customer's own active directory where they use their own accounts to authenticate against their own systems in their network. That sends a token across to LoadSpring, then we randomly generate a 64-character password that is only valid for that particular session. This process makes a successful brute force attack extremely difficult against a federated authentication environment.

- Lastly, we use **direct access prevention** where attackers must know the specific website that they are targeting versus being able to scan random IPs while guessing usernames and passwords.

Distributed Denial of Service (DDoS)



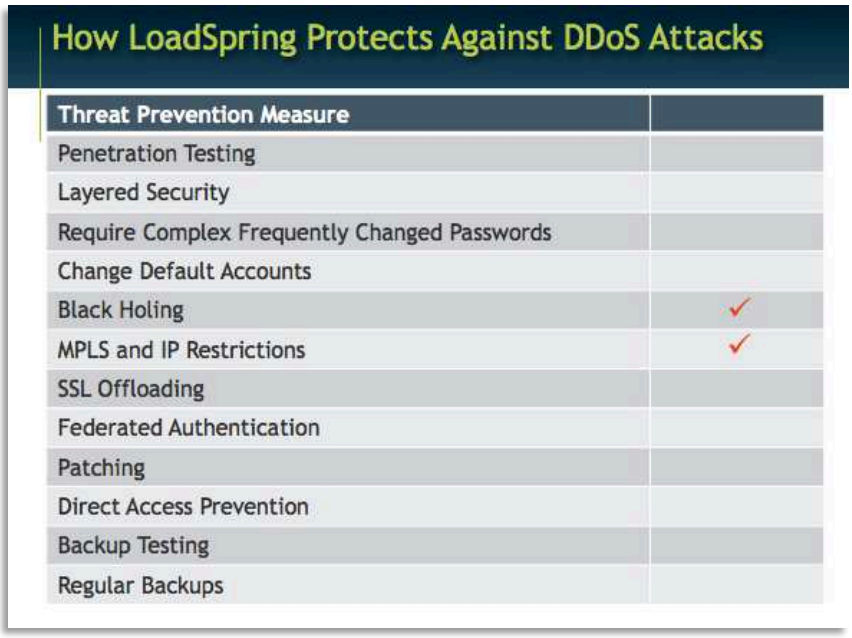
A DDoS is an attack where systems are compromised in an attempt to take an online service offline, or to prevent access by overwhelming the available bandwidth. This type of attack could happen directly against a data center, or against a critical service like your domain name system (DNS) provider. Your DNS provides visitors with access to your website or SaaS, so if an attacker is able to take it offline, legitimate users won't be provided access.

The risk with a DDoS attack is more on the access side and has gotten worse with the rise of the Internet of Things (IoT). IoT is a network of physical devices that enables them to connect and exchange data. An example would be connecting your light switch to the internet and controlling it through your smartphone. Controlling your home remotely sounds very efficient, the challenge is that most of those devices are not secured against a DDoS attack, and if compromised can become "zombies." If attackers put enough zombies together they can create a bot army allowing them to use the combined bandwidth of the compromised devices and use that to push data into or against a data center.

The DreamHost incident is an example of a successful DDoS attack, particularly during the controversy that surged after the Charlottesville, VA riots where they hosted a racist website.

Another successful attack was against Amazon Web Services (AWS) and one of their primary DNS providers. This attack impacted their hosted applications, which were taken offline for a period of four to eight hours.

How does LoadSpring protect against DDoS attacks?



How LoadSpring Protects Against DDoS Attacks

Threat Prevention Measure	
Penetration Testing	
Layered Security	
Require Complex Frequently Changed Passwords	
Change Default Accounts	
Black Holing	✓
MPLS and IP Restrictions	✓
SSL Offloading	
Federated Authentication	
Patching	
Direct Access Prevention	
Backup Testing	
Regular Backups	

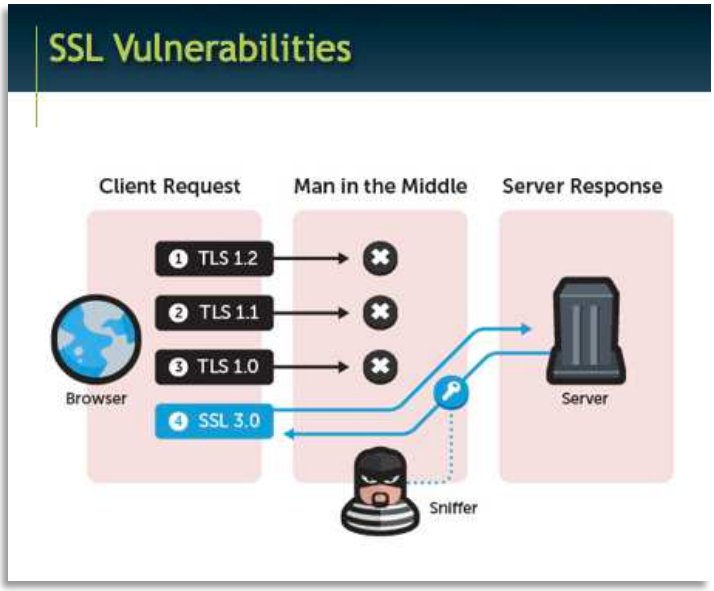
- We protect through a partnership with our internet service providers (ISPs). We have multiple devices in place that automatically detect if there is high traffic coming from an area that is not a legitimate user. This partnership enables us to redirect attacker traffic to a **black hole** or a null route before it saturates our bandwidth.

- **MPLS** is another measure we use against DDoS attacks being that it is a dedicated network independent from the internet attacks that go against primary websites. When we've seen DDoS attack attempts take place, customers that had MPLS were completely unaffected and remained online throughout the duration of those attacks. If a DDoS attack is a concern, the best way to protect against it is with a direct MPLS connection.

LoadSpring has not been targeted directly, but we have been impacted by DDoS attacks against someone using our same ISP or DNS service. When that happens, we are able to quickly work with our vendors to mitigate risk and limit the duration of the attack to under 30 minutes.

Secure Sockets Layer (SSL) Vulnerability

An SSL vulnerability attack exploits older SSL versions and/or vulnerabilities in current versions of transport layer security (TLS), where, if compromised, it allows attackers to pose as the legitimate website through a man-in-the-middle attack. If successfully exploited, an SSL vulnerability provides attackers with authentication information and enables them to gather information passively.



This attack is more unique to LoadSpring's hosting style, since we may be required to host older applications or versions that have outdated patches and SSL versions. We see this often with Oracle Contract Management, Primavera P6, and some of their older versions. When using an older software version and coming across a Heartbleed security bug that compromises the integrity of the SSL certificate, the attacker can pose as a legitimate website, then capture the information and credentials that pass back and forth between the website and the users. Through an SSL vulnerability attackers can obtain all of the session data, then reconstruct the information input by the user to give them a competitive image and access to your systems.

How does LoadSpring protect against SSL Vulnerabilities?

How LoadSpring Protects Against SSL Vulnerabilities

Threat Prevention Measure	
Penetration Testing	✓
Layered Security	
Require Complex Frequently Changed Passwords	
Change Default Accounts	
Black Holing	
MPLS and IP Restrictions	✓
SSL Offloading	✓
Federated Authentication	✓
Patching	✓
Direct Access Prevention	
Backup Testing	
Regular Backups	

- **Penetration testing**

- **MPLS restrictions** significantly reduce the likelihood of a man-in-the-middle incident since this measure keeps attackers from executing.

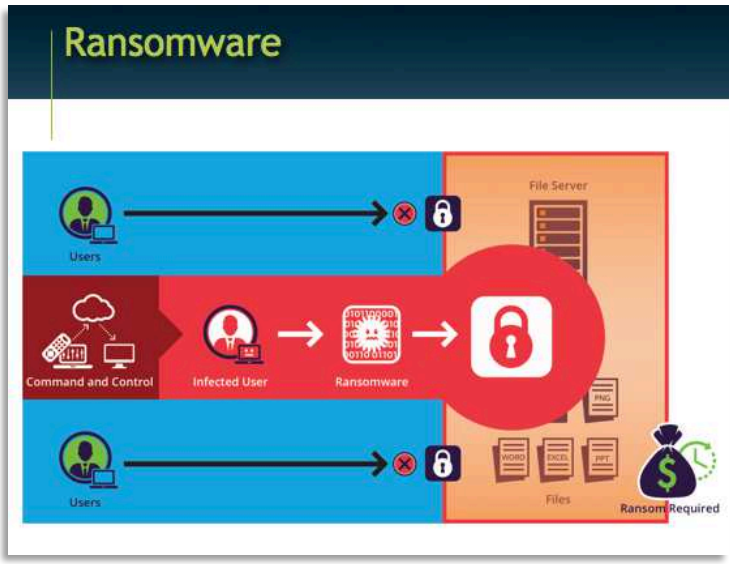
- With our **SSL offloading** we place an additional layer of defense along with a stronger certificate containing only the TLS versions allowed in the middle. This re-encrypts traffic with a stronger encryption, which allows us to circumvent some of the challenges of hosting older applications.

- **Federated authentication** prohibits attackers to capture session credentials due to the secure token.

- If a **patch** is available, then we patch as quickly as possible. Doing so allows us to minimize risk for older application versions with Heartbleed vulnerabilities.

Ransomware

Ransomware is a type of malicious software that has gotten significant media exposure lately. It exploits a website's vulnerabilities and encrypts all systems granting access to the owner only after a ransom is paid.



Oracle, one of the leading Engineering and Project Management applications providers, released over 270 patches during Q3, 2017 alone. Some of them were critical to block attackers from executing code remotely against their websites.

A ransomware attack took place against the San Francisco Municipal Transportation Agency (SFMTA) through a Primavera P6 EPPM vulnerability specific to this app.

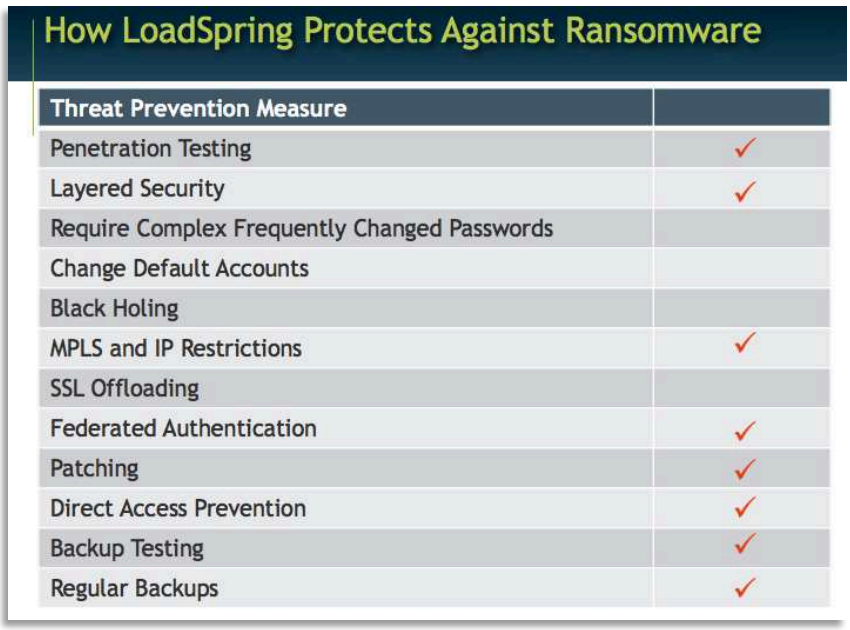
A successful ransomware attack encrypts all data and blocks accessibility. Even when backups that are tested and validated are in place, it can still take between 33 and 72 hours to recover from a ransomware attack.

Some victims of this attack pay the ransom, which could cost as much as $1,000 to $75,000 or more, in order to get the keys that should unlock your data.

Another victim of a ransomware attack through P6 was the Construction and Engineering company Rudolph Libbe, Inc., which paid a $30,000 ransom, still taking them three days to get their systems back up and running.

Symantec, the global leader in next-generation cyber security saw over 460,000 ransomware attacks in 2017 alone, and this number has grown over 30% as attackers continue to successfully obtain payment making ransomware attacks a bigger risk and threat.

How does LoadSpring protect against ransomware attacks?



How LoadSpring Protects Against Ransomware

Threat Prevention Measure	
Penetration Testing	✓
Layered Security	✓
Require Complex Frequently Changed Passwords	
Change Default Accounts	
Black Holing	
MPLS and IP Restrictions	✓
SSL Offloading	
Federated Authentication	✓
Patching	✓
Direct Access Prevention	✓
Backup Testing	✓
Regular Backups	✓

Ransomware hits across multiple areas, so we execute more prevention measures to ensure the integrity of our customer data remains intact.

- **Penetration testing**, since it discloses vulnerabilities or weaknesses in the software.
- Our **layered security** comes into play since we separate vulnerable web applications from your data using firewalls. So, if there is an attempted attack on a web server we can quickly and easily roll back to the latest snapshot of that server and experience no data loss being that no data is stored in the attacked server.
- **MPLS and IP restrictions** prevent attackers from accessing vulnerable applications.
- **Federated authentication** in combination with **direct access prevention** denies access to the protected apps until the user authenticates successfully. In these cases, attackers are unable to run scripts unless they have a valid user account.
- **Patching** also is critical when protecting against a ransomware attack. It is important to have a patch in place if a possible threat is detected during a penetration test.
- Lastly, **backup testing** and **regular backups**. It is very important to ask the business owners "how much data can they afford to lose?" being that many backup schedules are performed nightly, or every eight hours, when in reality the business owner can't afford to lose more than 15 minutes' worth of data. In the event of a direct ransomware attack, LoadSpring has backups in place that occur every 15 minutes to make sure that's the most amount of data we would lose, which enables our customers to quickly roll back.

One of the challenges we see with a ransomware attack is that sometimes the payloads can infiltrate a system and remain dormant for up to a year before they are activated. One of the ways we prevent that is by disallowing the outbound access from any of our servers. This means, if a payload got in it wouldn't be able to phone home and notify the attacker that it successfully penetrated our network. This prevents the attacker from remotely executing or activating that particular payload.

If we were to be impacted by a ransomware attack, one of the first things we would do is quickly isolate that particular box off the network, so that it can't infect anything around it. Then, we roll that server back to the time period before the infection took place. Since that server does not have customer data on it we can quickly restore access, identify how that vulnerability got in, then put a layer of security in place. This security measure can be in the form of a patch to the server, or restrict direct access by coordinating an IP restriction with the customer where we only allow application access for their worksites and not to the general internet.

If someone is individually infected at home, we recommend having good offline backups, as they enable the recovery of critical information. All users have to do is wipe the computer, patch it, get it up to date, then restore critical data using an offline backup.

Most of the attacks we see internally are not targeted specifically against LoadSpring, but are only random scans against our IP address ranges looking for vulnerabilities or exploits. In doing so, attackers look for things like: Have you exposed any of your database servers to the internet? Or, do you have any FTP servers or SFTP servers?

It is critical that companies stay up to date with patches. We often see companies get hit or attacked when patches have been out for a number of months, but their system admin teams or the security teams did not execute them.

When it comes to software, the more complex the application the more surface area there is for an attack against it. The enterprise applications we host are very complex, so our teams constantly go through and evaluate threats to ensure all the applications we host are protected regardless of the threat. For example, we identified a critical patch release for Oracle Identity Manager (OIM) that prohibits attackers from remotely executing code unless they authenticate to the OIM service, so we then evaluate whether there are other applications that leverage some of the same structure that could benefit from applying this patch on the web logic server.