

Injeção de SQL - Detecção de evasão

Resumo

A detecção dos ataques de injeção de SQL era feita inicialmente com o uso de técnicas de reconhecimento de padrões, verificados contra assinaturas e palavras-chave reconhecidamente maliciosas. Até pouco tempo, essa técnica foi bem-sucedida. Agora, os agressores escondem suas intenções malignas de várias formas, para evitar a detecção.

Essas tentativas de burlar a detecção exigem novas técnicas e tecnologias a fim de que sejam descobertas e neutralizadas antes que alcancem sistemas críticos e causem a exposição ou destruição de dados corporativos.

Os sistemas de detecção de evasão são uma nova forma de proteção contra tais ataques. Esses sistemas reconhecem as tentativas de disfarçar o código malicioso, que poderiam resultar em um ataque de injeção de SQL bem-sucedido, resultando na detecção e eliminação de ataques de injeção de SQL disfarçados.

O cavalo de Tróia

Historicamente, os ataques de injeção de SQL foram impedidos com o uso de técnicas de reconhecimento de padrões, armazenados em assinaturas em listas de palavras-chave para identificar solicitações potencialmente maliciosas. Como o cavalo de Tróia da lenda grega, o código perigoso está escondido em solicitações válidas e somente se torna aparente depois de aceito dentro dos muros do data center.

Os sistemas de prevenção contra ameaças, como o IPS e firewalls de software, podem lidar com essa técnica de invasão e são capazes de detectar esses ataques ocultos antes que eles sejam aceitos no data center e passem pela infra-estrutura de aplicativos, causando danos. Ao longo dos anos, foram compiladas listas enormes das combinações possíveis de palavras-chave e caracteres que podem resultar em um ataque de injeção de SQL, levando à criação de bases de dados de assinaturas. Como cada base de dados tem sua própria implementação do SQL, a linguagem de pesquisa de bases de dados padronizada do setor, essas assinaturas freqüentemente incluem ataques específicos contra certas bases de dados para garantir a detecção do maior número possível de ataques de injeção de SQL.

Além das assinaturas, o reconhecimento de palavras-chave normalmente é usado para impedir a penetração nas defesas de aplicativo. Certas palavras-chave como DROP e UNION são freqüentemente usadas em ataques e, portanto, recusadas quando descobertas em solicitações destinadas a qualquer implementação de base de dados.

Os agressores descobriram que o "Cavalo de Tróia" é facilmente reconhecido pelos firewalls, e é impedido de penetrar na infra-estrutura de aplicativos. Eles tiveram de descobrir uma nova maneira de introduzir seus ataques nos data centers, e então nasceu a "Zebra de Tróia".

A zebra de Tróia

A zebra de Tróia é muito parecida com o cavalo de Tróia, usado antigamente para executar ataques, mas suas cores e padrões são diferentes. Os sistemas de prevenção contra ameaças, procurando por padrões específicos, são confundidos pela aparência da zebra de Tróia e permitem-lhe penetrar no data center, onde o seu ataque oculto é facilmente executado contra as bases de dados corporativas.



Os ataques de injeção de SQL são bem-sucedidos porque a zebra se parece muito com um cavalo, mas está disfarçada em vários padrões de listras que não estão presentes nas bases de dados de assinaturas de hoje. Pior ainda, os padrões mudam de zebra para zebra, tornando as técnicas de comparação de padrões virtualmente inúteis.

Os ataques de injeção de SQL hoje são como as listras na Zebra de Tróia - o perigo está lá, mas está escondido por uma variedade de padrões de listas, e virtualmente indetectável. Os agressores usam a flexibilidade de parâmetros de linha e a variedade de linguagens para ocultar ataques tradicionais contra os sistemas de prevenção contra ameaças. Assim como o padrão de listras da zebra é único, cada ataque de injeção de SQL executado com o uso de novas técnicas para disfarçar sua verdadeira intenção também é único.

Ataque tradicional de injeção de SQL	Técnica de evasão	Ataque oculto de injeção de SQL
...71985' OR '1'='1'	Manipulação de espaços em branco	...71985'OR'1'='1'
'&id=111 UNION /**/ SELECT *...'	Comentário - sintaxe C	&id=111/*Este é um comentário...*/UN/*Você consegue */IO/*encontrá-lo*/N/**/S/**/E/**/LE/*Outro comentário para*/CT/*encontrar. se você for*//*capaz*/
1 UNION SELECT ALL FROM WHERE	Codificação: Hexadecimal	1 UNION SELECT ALL FROM WHERE
	Codificação: BASE 64	MSBVTkiPTiBTRUxFQ1QgQUxMIEZST00gV0hFUkU=
	Codificação: DECIMAL	1 UNION SELECT ALL FROM WHERE
...71985' OR '1'='1'	Variações de um tema	...71985' OR 'cidade' = 'muzambinho'

Essencialmente, hoje há quatro categorias de técnicas de evasão em uso.

Manipulação de espaços em branco

Quase todos os sistemas modernos de detecção de injeção baseados em assinaturas são capazes de detectar ataques que variam o número e a codificação de espaços em branco em torno do código SQL malicioso. O que esses sistemas não podem perceber é a falta de espaços em branco em torno do mesmo código. Como os métodos de verificação de padrões e assinaturas procuram por padrões de texto que incluem um ou mais espaços, não conseguem detectar o mesmo padrão de texto quando nenhum espaço é incluído.

Note também que, embora a manipulação de espaços em branco normalmente envolva a remoção de espaços em branco de uma pesquisa em potencial, os seguintes caracteres também podem ser usadas para confundir os sistemas de detecção baseados em análise de padrões e assinaturas:

- Tabulação
- Retorno de carro
- Nova linha

Essa evasão funciona porque o sistema de análise de SQL da maioria das bases de dados não leva em conta os espaços em branco e os caracteres de formação, e na verdade são escritos para atender a uma quantidade variável desses caracteres em torno de palavras-chave.

Exploração de comentários

No passado, os agressores usavam a sintaxe de comentário de hífen duplo, por exemplo, `--`, para mascarar suas intenções. Muitos sistemas hoje podem detectar essa tentativa e por isso os atacantes mudaram de tática, usando a sintaxe de comentário da linguagem C, por exemplo, `/* */`, para evitar a detecção.

Os atacantes usam os comentários no estilo C no lugar de espaços, para separar comandos que são normalmente usados juntos em ataques, mas são facilmente detectados mediante a verificação nas assinaturas. Em alguns casos, são usados para separar palavras-chave.

Por exemplo, UNION é uma palavra-chave comum usada em ataques de injeção de SQL. Embora os sistemas de detecção baseados em assinatura nem sempre marquem a palavra-chave UNION como um ataque potencial, quando não acompanhada de um espaço ou outra palavra-chave conhecida, uma verificação de palavras-chave irá marcar UNION como um ataque em potencial. Para evitar essa detecção, os atacantes podem tentar usar os marcadores de comentário no estilo C para dividir a palavra-chave, por exemplo, `UN/**/ION`, dessa forma derrotando não somente a verificação de assinaturas, mas também a de palavras-chave.

Esse ataque funciona porque a maioria dos sistemas de análise SQL das bases de dados modernas removem todos os comentários das pesquisas antes de processar o pedido SQL (tudo entre `/*` e `*/`), e o que resta então é uma declaração SQL perfeitamente válida.

Técnicas de codificação

As técnicas de codificação provavelmente são o método mais fácil de derrotar os sistemas de detecção de padrões e assinaturas. Isso acontece porque a codificação tem o efeito de mudar completamente o texto, da mesma forma que a criptografia muda o texto para escondê-lo das pessoas que não devem lê-lo.

As codificações mais comuns usadas para invadir a detecção são:

- Codificação de URL
- Unicode/UTF-8
- Codificação hexadecimal
- Função `char()`

As técnicas de codificação funcionam em função da natureza heterogênea da web, da necessidade de suportar inúmeros idiomas e conjuntos de caracteres e da incapacidade das soluções de prevenção contra ameaças de decodificar adequadamente ou oferecer suporte a múltiplas páginas de código para solicitações de entradas, antes da aplicação de métodos de detecção de ameaças.

Variações de um tema

Há múltiplas variações de disfarce de ataques que usam as capacidades originais da linguagem SQL, conforme definidas no padrão SQL99.

Concatenação

A concatenação divide palavras-chave identificáveis e evita a detecção explorando a capacidade original do SQL de construir uma linha única a partir de várias partes. A sintaxe da concatenação varia de acordo com a base de dados, mas, de maneira geral, usa os caracteres mais (+) e a barra vertical (||) para indicar a concatenação ao sistema SQL.

Por exemplo:

```
EXEC('SEL' + 'ECT US' + 'ER')  
EXEC('SEL' || 'ECT US' || 'ER')
```

Conversão

As técnicas de conversão fazem uso da capacidade original do sistema SQL de converter tipos de dados. Isso permite ao agressor evitar a detecção introduzindo funções SQL válidas que mudam a assinatura de uma declaração.

Por exemplo:

```
OR username = char(37) /* 37 equivale ao caractere curinga do SQL, %*/
```

Variáveis

Muitos sistemas permitem a declaração de variáveis, que podem então ser usadas não apenas para contornar a detecção pelo software do firewall, mas também a validação adicional de entrada baseada em código.

Por exemplo:

```
; declare @myvar nvarchar(80); set @myvar = N'UNI' + N'ON SEL' + N'ECT U'  
+ N'SER'); EXEC(@myvar)
```

A zebra de Tróia consegue executar ataques porque essas técnicas de evasão resultam em um número impossivelmente alto de combinações em assinaturas. Em muitos casos, elas podem ser usadas em conjunto. Por exemplo, uma delas pode manipular espaços em branco na solicitação original e então codificar completamente toda a solicitação, exigindo não só uma, mas duas detenções bem-sucedidas para impedir o ataque. Esse súbito aumento nas assinaturas de ataque possíveis não pode ser combatido com o uso de técnicas tradicionais de análise de padrões ou bases de dados de assinaturas.

Uma nova técnica é necessária para detectar não somente o ataque, mas a tentativa de evitar a detecção.

O sistema de políticas de detecção de evasão

Para detectar com sucesso tanto o ataque de injeção de SQL quanto uma tentativa de detecção, é necessário incorporar tecnologias de detecção de invasão nas soluções de prevenção contra as ameaças existentes, como o BIG-IP® Application Security Manager (ASM).

O ASM agora inclui uma tecnologia sofisticada contra evasão, criada para detectar e neutralizar ataques evasivos de injeção de SQL. Essa tecnologia, chamada de Policy Evasion Detection Engine (imPEDE), é capaz de reconhecer vários métodos de evasão, impedindo-os de alcançar o alvo.

O imPEDE do ASM cumpre essa tarefa normalizando os dados que passariam pelos sistemas tradicionais de prevenção contra ameaças, os quais dependem de sistemas de verificação de padrões e assinaturas. Normalizando os dados, independentemente do formato em que eles chegam, o imPEDE consegue remover o impacto das tentativas de evasão na verificação de bases de dados de assinaturas e palavras-chave.

As técnicas de normalização do imPEDE funcionam porque os ataques de injeção de SQL, em si, não mudaram, mas as maneiras com que eles são embutidos nas solicitações. Ao detectar as tentativas de evitar a detecção pelo sistema subjacente, o imPEDE permite que os métodos comprovados do ASM de prevenção contra a injeção de SQL continuem protegendo os aplicativos e os dados armazenados nas bases de dados corporativas.

O imPEDE aumenta ainda mais a segurança sem afetar o desempenho - uma preocupação constante na implementação de sistemas de prevenção contra ameaças e firewalls de aplicativos web em geral - por meio do emprego da detecção baseada em políticas. O imPEDE permite, por meio de políticas, a determinação de quais URLs devem ser examinadas e quais devem ser consideradas como livres de ameaças. Normalmente, as políticas são aplicadas às URLs de envio de dados, mas não necessariamente àquelas que apenas recuperam e exibem dados, visto que essas últimas têm menor probabilidade de conter ameaças em potencial enviadas por agressores.

A abordagem flexível do imPEDE, baseada em políticas, permite ao administrador determinar o que deve ou não ser protegido, e pode ser melhorada ainda mais com a capacidade do ASM de monitorar e relatar as mudanças no site, que incluem novas URLs ou mudanças no comportamento das existentes. Isso permite aos administradores tomar decisões, independentemente do nível de segurança necessário, para cada URL, conforme o site muda, tornando a exploração baseada em sites muito mais fácil e simples.

Conclusão

Um sistema comum de prevenção contra ameaças, baseado em assinaturas ou padrões, como o IPS, não pode lidar corretamente com os ataques evasivos. Embora essas técnicas sejam uma boa base para impedir as ameaças conhecidas de alcançarem os aplicativos, elas constituem um método estático de detecção de ameaças, que não poderá acompanhar a evolução dinâmica dos ataques da web, particularmente o de injeção de SQL.

Tecnologias avançadas como o imPEDE do ASM são necessárias para deter as técnicas de evasão usadas para invadir alvos por meio das soluções de prevenção contra ameaças existentes. Essas soluções, como o IPS ou firewalls independentes de aplicativos web, oferecem proteção principalmente na camada de aplicação da web, e não podem lidar com o problema maior da segurança na distribuição de aplicativos. O ASM, combinado com a segurança nas camadas de rede e de transporte de aplicativos em uma plataforma de distribuição de aplicativos, e integrado a uma rede de distribuição de aplicativos, oferece uma solução holística para garantir a distribuição segura, rápida e disponível dos aplicativos.