POLYMORPHIC EM AMBIENTES OPEN SOURCE

Bloqueia ataque antes que este ocorra

TABLE OF CONTENTS

POLYVERSE

Resumo Executivo

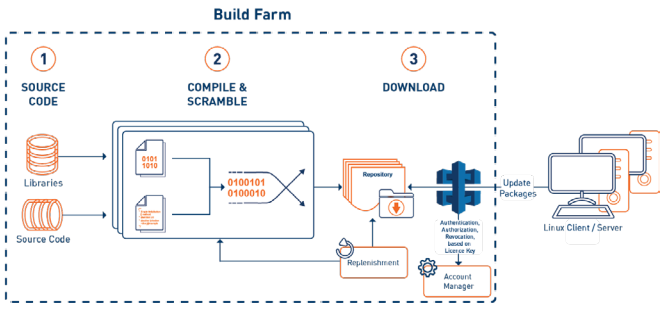
Ambientes críticos, empresas digitais, serviços públicos e militares trabalham incansavelmente para mitigar o risco de aumentar e adaptar rapidamente os ataques cibernéticos encadeados. Esses ambientes complexos estão sob constante ameaça para proteção propriedade intelectual, dados confidenciais e infraestrutura crítica etc. Com ambientes de produção altamente complexos baseados em tecnologia em código aberto com soluções de terceiros, essas não possuem visibilidade e o controle necessários para garantir que não ocorram violações e/ou invasões.

Na atualidade, os usuários de software contam com a integridade de funcionários, parceiros e na colaboração da comunidade código aberto para fornecimento de sistemas operacionais Linux confiáveis e livres de malware. No entanto, a possibilidade de invasores introduzirem malware no código-fonte ou no nível de distribuição é um risco real e que de fato é explorado por cyber criminosos. A violação da SolarWinds em 2020 trouxe a questão de "como posso garantir a segurança em minha empresa quando não consigo monitorar e controlar continuamente o software que está sendo usado?", tema que está na ordem do dia dos profissionais de segurança. Tal violação era inevitável e, talvez mais importante, as violações futuras da no encadeamento de eventos que a mesma têm de probabilidade de ocorrer novamente.

As soluções de software de confiança zero da Polyverse Corporation interrompem os ataques antes que eles comecem. O Polymorphic Build Farm para Open Source é uma solução turnkey para ambientes de DevSecOps e garante a integridade e a segurança no encadeamento de códigos-fonte. A empresa não precisa mais depender de distribuições públicas para baixar a versão mais recente de seus sistemas operacionais. Com a tecnologia de segurança de codificação binária da Polyverse, crie e personalize suas próprias distribuições e obtenha controle total das atualizações de código.

Polymorphic implementado em ambiente Open Source

Dado o risco real de backdoors de segurança no código-fonte aberto do Linux, a solução de software de confiança zero da Polyverse atenua esse vetor de ameaça. Embora "Linux" seja comumente usado como o nome de um sistema operacional inteiro, na verdade ele se refere ao kernel e ao software que gerencia as interações entre o hardware e os aplicativos do usuário final. Uma distribuição Linux, como SUSE, Red Hat, Ubuntu, Debian ou CentOS, refere-se a um sistema operacional completo construído sobre o kernel Linux. Uma distribuição é composta do kernel Linux e uma infinidade de pacotes que o fornecedor verificou para interoperar conforme as suas necessidades de serviços. Esses pacotes podem ser baixados e instalados individualmente usando um gerenciador de pacotes como yum, rpm, zypper ou apt. Ao lançar versões atualizadas de seu sabor do Linux (por exemplo, CentOS ou Ubuntu), os fornecedores reconstroem todos os pacotes. Para a maioria dos usuários, atualizar todos os pacotes durante a atualização é perfeitamente normal; entretando, em circunstâncias específicas, as organizações precisam controlar quais atualizações podem ser instaladas.



A Polyverse aplica os padrões de confiança zero do NIST ao software por meio do Polymorphic Build Farm for Open Source. Um Polymorphic Build Farm oferece controle total de seu ambiente, fornecendo um ambiente de construção de ponta a ponta para o ecossistema Linux que constrói todo o Linux com compiladores aprimorados, configurações personalizadas e modificações de código. Quando combinado com um compilador Polymorphing, o Polymorphic Build Farm pode produzir um número arbitrário de binários únicos que têm semântica e desempenho equivalentes (por ordem aleatória de função, tabelas PLT, blocos init / fini, alocação de registro etc.). Esses binários são extremamente difíceis de hackear e diminuem ou bloqueiam ameaças baseadas em memória conhecidas e não mapeadas.

Visão Geral dos Benefícios

O Polymorphic Build Farm para código aberto se integra perfeitamente aos processos existentes de DevOps e implantação de patch de TI. Oferecendo controle total de compilação de toda a pilha, do kernel às bibliotecas, a tecnologia fornece uma base poderosa para inovação. O código e os binários resultantes em um Polymorphic Build Farm podem ser modificados para qualquer necessidade.

Polymorphic Builds

The Polymorphic Build Farm enables enhanced security on modO Polymorphic Build permite maior segurança em sistemas modernos e legados. No núcleo está o conceito de um compilador substituível para o processo de construção. O compilador da Polyverse, Polymorphic compiler mistura e embaralha um pacote Linux de modo que cada repositório compilado tenha uma representação binária exclusiva que possa ser usada. O resultado é uma versão completamente única de seu sistema operacional em nível binário que funciona, executa e opera exatamente da mesma maneira, mas é imune a ameaças baseadas em memória e sem arquivo,

que representam 80% dos ataques cibernéticos bem-sucedidos.

Sistema de Distribuição
O sistema de distribuição Polymorphic rastreia cada host que solicita um repositório Linux. Depois que um repositório é alocado para um host, apenas esse host pode fazer download de pacotes. Os administradores podem determinar facilmente a partir do painel qual host Linux solicitou e baixou cada repositório correspondente. Este ambiente de construção cria uma versão que se integra com sistemas de controle de versão como GIT para que cada linha de código possa ser inspecionada, verificada e, se necessário, revertida.

Cadeia de Integridade
Ao compilar a partir de seu próprio código-fonte, pode-se realizar inspeções e varreduras de código antes de entregar compilações aos hosts da sua organização. O download de código-fonte aberto ou binários traz o risco de potencialmente baixar um malware embutido no código ou de um ambiente espelhado (mirror) que redireciona o host para baixar pacotes de fontes não confiáveis. Não obstante, os usuários podem presumir que outra pessoa na comunidade verificou a integridade do código, todavia é possível que ninguém tenha verificado ou atestado que as verificações de segurança foram realizadas, parte disso cria uma falsa impressão de segurança.

Point-in-Time Cache
Com o Polymorphic, pode-se congelar uma versão específica de uma distribuição no tempo (por exemplo, Ubuntu versão 16.04). Atualizações e patches individuais podem ser aplicados manualmente para pacotes específicos, conforme necessário. Isso permite que haja um controle em um nível muito granular as compilações do Linux geradas pelo Polymorphic Build Farm.

Considerações de Segurança
As compilações do Linux podem ser instrumentadas para proteger contra a ataques de malware sem arquivo estáticos e dinâmicos. As contramedidas são adicionadas no processo de construção, incluindo:

- Embaralhamento da estrutura de dados - injete bytes de preenchimento automaticamente, reordem de campos ou introduzação a outro embaralhamento de estruturas de dados para verificar o acesso não conforme
- Dynamic stack canary – alteração de pilha antes de alocação de endereço de retorno para evitar que os invasores adivinhem por meio de força bruta esses dados
- Shadow stack - armazene os endereços de retorno em uma pilha separada para garantir a integridade do fluxo de controle, validando que os endereços de retorno não foram adulterados antes da execução
- Page Load Table (PLT) randomization – Mova a PLT durante a execução do processo para evitar a descoberta por invasores e validar os chamadores (não uma chamada de salto aleatória)
- Code motion – movimente de funções de biblioteca dinamicamente durante a execução (Q4 2021)

Injeção automática de controles de segurança
Um caso de uso muito comum para o Polymorphic Build Farm é injetar automaticamente controles de segurança adicionais nos sistemas. Esses controles podem variar de simplesmente adicionar verificação de integridade de fluxo de controle a uma criptografia muito mais sofisticada na memória de estruturas de dados que gira automaticamente a criptografia.

Configuação automártica de tesate
Mesmo depois de décadas de pesquisa, o modelo de testes sobre bugs em softwares continua sendo um tema desafiador. Condições de limite, condições de corrida e outros cenários podem ser particularmente difíceis de solucionar e resolver. Há uma razão para que sistemas complexos como o Microsoft Windows funcionem bem logo após uma reinicialização, mas decaem gradualmente com o tempo. Os primeiros cinco minutos são mais fáceis de testar e depurar do que um sistema que está em execução há um mês.

O Polymorphic Build Farm pode melhorar significativamente a facilidade de teste do sistema. Com o Polymorphic Build Farm, assim como várias variações de sua pilha de software podem ser criadas para fins de segurança, pode-se criar várias variações do Linux para fins de teste. Alterações de atraso de tempo, alterações de layout de memória, ordenação de estrutura de dados e muito mais podem ser modificados de maneiras muito sistemáticas e completas para cobrir mais amplamente uma variedade de cenários de teste nunca antes realizados. Como um exemplo simples, as chaves de contexto de thread podem ser chamadas automaticamente antes, durante e depois do código de tipo de seção crítica para ajudar a testar se essas construções foram implementadas corretamente.

Como contrstruir o ambiente operacional

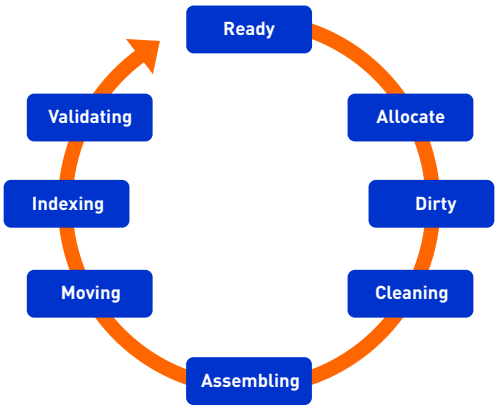
O Polymorphic Build Farm é um dos maiores builds do mundo reconstruindo o Linux, gerando quantos repositórios forem necessários com grande escalabilidade. Polyverse atualmente oferece suporte a mais de 25 distribuições e versões, incluindo versões e configurações personalizadas. Para cada distribuição Linux, o Polymorphic Build Farm gera vários repositórios de cada distribuição. Quando um host Linux solicita atualizações do Polymorphic Build Farm, aloca-se um repositório para esse host

para uso exclusivo. O host apenas baixa pacotes desse repositório específico, o que atende às necessidades desse host.

Além de gerar repositórios exclusivos por máquina, hospeda-se a sua própria versão do compilador Polymorphing, usando o compilador Polymorphing, há condições para que se baixe pacotes codificados exclusivamente em cada download. Tal como acontece com o produto Polymorphing regular da Polyverse, todos os pacotes especificados teriam um layout de conjunto de instruções exclusivamente embaralhado no nível binário. Junto com segurança adicional e benefícios de teste, o Polymorphic Build Farm também fornece funcionalidade aprimorada. Seu host pode baixar perfeitamente pacotes aprimorados - em todos os casos, usando a atualização padrão e os mecanismos de correção já fornecidos nas distribuições do Linux.

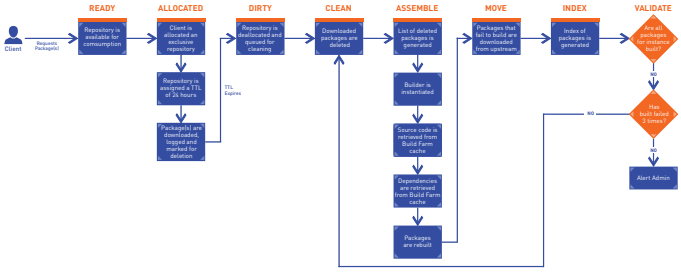
Por padrão, o Polymorphic Build Farm verifica os repositórios públicos em busca de atualizações a cada hora e baixa todas as alterações. Ao contrário dos pacotes Linux fornecidos pelo fornecedor, um Polymorphic Build Farm permite que você crie um cache point-in-time configurável para interromper o download de atualizações para uma determinada distribuição e versão (por exemplo, Ubuntu versão 16.04). Você pode proteger seu software legado e as versões do Linux que não estão mais disponíveis na própria distribuição original. Com esta configuração customizada, o Polymorphic Build Farm construí não apenas novos repositórios para a distribuição Linux, mas verifica a integridade que seja especificada. As atualizações adicionais individuais podem ser sincronizadas manualmente para determinados pacotes, conforme a empresa considerar adequado ou baseado na sua demand.

O Polymorphic Build Farm é composto por oito estados, com serviços gerenciando cada estado individual. A figura a seguir mostra esses estados.



O diagrama de estado a seguir ilustra o processo lógico que o Polymorphic Build Farm executa quando uma cliente baixa um pacote.

Quando um repositório está no estado “pronto”, ele se torna disponível para download. Quando um cliente solicita um pacote, o Polymorphic Build Farm aloca um repositório para o cliente para uso exclusivo. O repositório recebe um tempo de expiração (TTL Time to Live) de 24 horas. Depois que o TTL expira, o repositório é automaticamente marcado para exclusão. O cliente pode baixar pacotes deste repositório quantas vezes desejar dentro do período de 24 horas. Cada download é registrado para fins de rastreamento. Após a expiração do repositório, se o usuário solicitar um pacote, o Polymorphic Build Farm aloca outro repositório para o usuário.



Uma vez que o TTL do repositório expira, o mesmo é desalocado e enfileirado para limpeza. O Polymorphic Build Farm inspeciona o repositório para determinar quais pacotes foram consumidos pelo cliente e os elimina do repositório. O Polymorphic Build Farm gera um manifesto de pacotes excluídos para reconstruir e as configurações necessárias para cada pacote. Um contêiner docker, chamado Builder, é gerado com as dependências e o código-fonte necessários para reconstruir os pacotes embaralhados. Uma vez que os pacotes são construídos, eles são validados, indexados e prontos para consumo pelos clientes. Se algum dos pacotes falhar na reconstrução, o Polymorphic Build Farm tenta, antes de alertar o administrador, reconstruí-los até três vezes e baixá-los de um repositório público upstream.

Conclusão

Os ataques baseados em encadeamento de software são notoriamente difíceis de resolver quando uma pilha de software contém software de código aberto e de fornecedores terceirizados, além de quaisquer aplicativos personalizados desenvolvidos internamente. A solução é colocar o controle de volta nas mãos de sua empresa. Ao construir toda a pilha de software, não apenas seu próprio aplicativo, você obtém controle completo por meio do gerenciamento de configuração centralizado e, mais importante, visibilidade do que está sendo executado. Este é efetivamente um processo DevOps de pilha completa, permitindo que você tenha os mesmos padrões para toda a pilha que já possui para seu próprio código.

Um Polymorphic Build Farm para Open Source dá à sua organização integridade do código-fonte com a capacidade de alterar e

embaralhar qualquer distribuição Linux dentro de seus próprios ambientes. Isso garante que nenhum backdoors ou malware seja introduzido, fornece a capacidade de bloquear sua configuração padrão do Linux e protege sua imagem do Linux antes da instalação. A solução oferece a oportunidade de criar caches point-in-time ou realizar atualizações parciais. Gerenciando seu repositório de código-fonte, você pode garantir melhor a segurança de seus ambientes de TI legados e de missão crítica.

Ao empregar essa tecnologia, o controle de segurança retorna as empresas, dando-as a executação segura em seus ambientes computacionais. Com um Polymorphic Build Farm, os cyber criminosos não possuem mais a vantagem de usar pontos de vulnerabilidades e e backdoors. O bloqueio de ataques antes que estes comecem, de fato é uma soluções que ratifica o conceiro de Zero Trust da Polyverse

For More Information

Contact us at:

sales-us@polyverse.com

sales-emea@polyverse.com

sales-apac@polyverse.com

Or visit our website:

https://polyverse.com